I P K S

Prozess - Software - Entwicklungs GmbH

M2000 + SICOMP M/R Emulation

for Windows

User Manual

Limitation of Liabilities

While every precaution has been taken in the preparation of this book, the publisher assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

Right to Use

The contract conditions for the use of IPKS software by the licensee are submitted for perusal and approval prior to the installation of the software.

Publisher

IPKS Prozess-Software-Entwicklungs GmbH

Am Weichselgarten 36
91058 Erlangen
Tel.: +49 (0) 9131/69588-0
Fax.: +49 (0) 9131/69588-25
E-mail: support@ipks.de

Edition

09 January 2007

Program Version

The manual describes the software products M2000 (version 4 and higher) and SICOMP M/R Emulation.

Trademarks

Trademarks or patents generally protect all references to software or hardware used in this document.

Contents

1 Introduction 4

2 Functional Description 7

2.1 Software Structure 7
2.2 Scope of Performance 8
2.2.1 CPU Emulation 8
2.2.2 Device Emulation 8

3 Commissioning 16

3.1 Scope of Delivery 16
3.2 System Requirements 19
3.3 Installation 20
3.3.1 Overview 20
3.3.2 Installing M2000 (incl. Emulator) 22
3.3.2.1 Installing the User System 23
3.3.2.2 Changed System Options 23
3.3.3 Configuration File (mpar.sys/rpar.sys) 24
3.4 Emulator Operation with M2000 29

4 Configuration File for SIC-M 30

4.1 General Entries 30
4.2 Device Definition Entries 32
4.2.1 Visual Display Units 32
4.2.2 Storage Media 38
4.2.3 Printers 41
4.2.4 Interface Connections 49
4.2.5 Additional Devices 58
4.2.5.1 Time Synchronization 58
4.2.5.2 Timer 59
4.2.5.3 Profibus & Pseudo Devices 60
4.2.5.4 Promea MX & BDE Terminal (ES 1) 65

5 Configuration File for SIC-R 67

5.1 General Entries 67
5.1.1 Visual Display Units 69
5.1.2 Storage Media 73
5.1.3 Printers 76
5.1.4 Interface Connections 83
5.1.5 Additional Devices 89

6 Device Emulation Details 96

6.1 DS074 Device Emulation 96
6.1.1 Screen Structure 96
6.1.2 Key Assignment 97
6.2 DU05 Computer Interfacing 99
6.2.1 DU05 Emulation via DF42 99
6.2.1.1 Requirements 99
6.2.1.2 Parameter Definition 99
6.2.1.3 MSV2 Protocol Parameters 100
6.2.2 Diverting DU05 Communication to TCP/IP 100
6.3 DU06 Computer Interfacing 101
6.3.1 DU06 Emulation via DF32/42 101
6.3.1.1 Requirements 101
6.3.1.2 Parameter Definition 102
6.3.1.3 HDLC-UNC Protocol Parameters 103

6.3.1.4 HDLC-BAC Protocol Parameters 103
6.4 Open DU0x Communication Interface 104
6.4.1 Overview 104
6.4.2 Diverting DU04 Communication to TCP/IP 104
6.4.3 Diverting DU05 Communication to TCP/IP 104
6.4.4 Diverting DU06 Communication to TCP/IP 106
6.5 CP1400 Communication Control 107
6.5.1 Installation 107
6.6 CS275 Communication Control 108
6.6.1 Module Installation 108
6.6.2 Software Installation 108
6.7 DUST3965R Computer Interface Connection 109
6.7.1 DUST3965R Emulation via DF42 109
6.7.1.1 Requirements 109
6.7.1.2 Parameter Definition 109
6.7.1.3 MSV2 Protocol Parameters 110
6.8 DUST3961 Computer Interface Connection, Gateway Function 111
6.9 DUST 3966 Computer Interface Connection 112
6.9.1 DUST 3966 Emulation via DF32/42 112
6.9.1.1 Requirements 112
6.9.1.2 Parameter Definition 112
6.9.1.3 HDLC-UNC Protocol Parameters 113
6.9.1.4 HDLC-BAC Protocol Parameters 114
6.9.2 Diverting DUST 3966 Communication to TCP/IP 114
6.9.3 Diverting DUST3964R Communication to TCP/IP 114
6.10 PE3600 Emulation via ProfiBus or SFT Module 115
6.10.1 M2000 115
6.10.2 Configuration File *rpar.sys* 115
6.10.3 Example 118
6.11 PE3600 – Siemens SFT Module 118
6.11.1 Example 118
6.12 Interface Concentrator 119
6.12.1 General 119
6.12.2 Parameters in *mpar.sys* 119
6.12.3 Proxy Processes on the Remote Computer 120
6.13 OP31 Emulation 121
6.13.1 Setup Instructions for SONY RMO-S580 Drive 121
6.13.1.1 Function Switch on the Rear Panel 121
6.13.1.2 Commissioning 121
6.13.2 Important Notes 122
6.13.3 Data Media 122
6.14 INOPERABLE Detection for Serial Interfaces 123
6.14.1 TTY / V.24 Interfaces 123
6.14.2 V.24 Connection 123
6.15 Compatibility Restrictions 124
6.15.1 SICOMP M Computer 124
6.15.2 SICOMP R Computer 126
6.16 Communication Channel SCU – SICOMP M/R Programs 127

7 Auxiliary Programs 128

7.1 The PCCOPY Program 128
7.2 The MCSAVE.EXE Program 129
7.3 The WAIT4.EXE Program 130
7.4 The SOLDIS Program 131
7.5 SicView, M2000 Hard Disk Explorer 132

8 PE F7 Emulation 133

8.1 Hardware Structure 133

8.1.1 SICOMP M 133

8.1.2 M2000 134

8.2 Software Structure 135

8.2.1 Access to the Process I/O Equipment 136

8.2.2 Alarm Processing 136

8.3 Functional Description 137

8.3.1 Device Emulation 137

8.3.1.1 Element Calls 137

8.3.1.2 Device Function 137

8.3.1.3 Addressing 137

8.4 ProfiBus DLL 138

8.4.1 General 138

8.4.2 Installation (ProfiBus) 138

8.5 Parameter Definition 139

8.5.1 Configuration File *mpar.sys* 139

8.5.2 Example (ProfiBus) 140

9 Appendix 141

9.1 General Notes 141

9.2 Messages 142

9.2.1 Start, End and Error Messages 142

9.2.2 Trace Messages 142

9.2.2.1 TrcView Window 142

9.2.2.2 TrcView Features 142

9.3 Template of the Parameter File *mpar.sys* 143

9.4 Template of the Parameter File *rpar.sys* 149

9.4.1 Template of the StartBatch *emu_strt.bat* 152

9.5 M2000 Error Notification Form 153

9.6 Error Analysis by IPKS GmbH 154

10 Index 155

1 Introduction

In the past decades, Siemens SICOMP M/R minicomputers have demonstrated their flexibility, high degree of availability and powerful system software in numerous industrial and commercial applications worldwide. Over the period, users have invested substantial financial and engineering expenditure in extensive complex application software especially designed for these proprietary computers. These dedicated software systems have over many years proven their usefulness and stability in mission critical applications, and in many cases their functional scope is sufficient for years to come.

However, due to the enormous progress made in hardware development over the past years, personal computer technologies have evolved into an internationally recognized hardware standard that has found solid ground even in industrial applications.

Although the Siemens SICOMP M/R hardware is not able to benefit from this technological progress, you want to open your SICOMP M/R system to today's computer world and benefit from advantages such as:

- Make use of cost-effective PC components
- Profit from latest developments in hardware components
- Reduce maintenance costs
- Integrate your system in the Windows world

M2000 is our solution for you. With the one-to-one emulation of the SICOMP M/R hardware command set and the SICOMP M/R level structure, M2000 maps your SICOMP applications to the command set of the Intel x86 processors (or to compatible processors). This enables you to run the software generated on a SICOMP M/R computer on a Windows computer.

Figure 1 compares the original Siemens SICOMP M/R system with the emulated system running on a computer based on Windows and M2000. The M2000 emulator has the task of adapting the SICOMP M/R command set to the Intel command set, and of connecting the original I/O devices to the Windows environment.

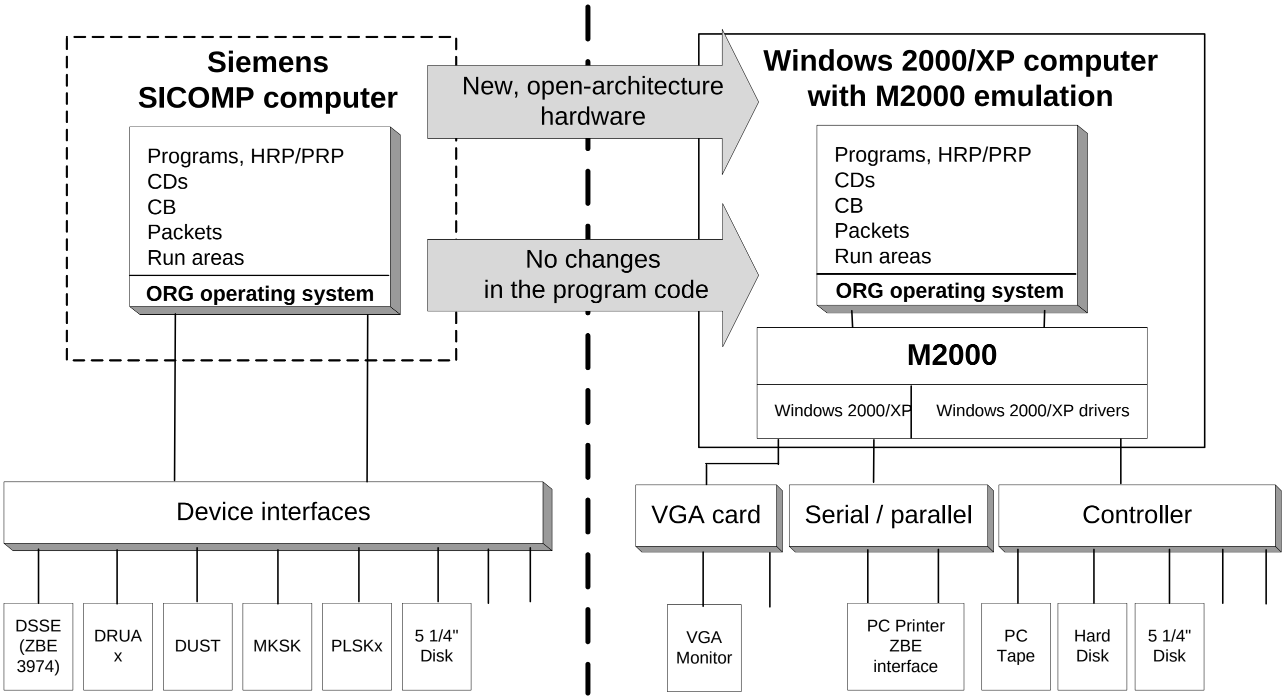


Figure 1: Comparison SICOMP - M2000

Input and output requests issued by the ORG-M/ORG-R operating system and by Siemens SICOMP programs are processed by the I/O drivers of the M2000 emulator. I/O requests addressed to disk memories (i.e., PLSK) are redirected to Windows files and the PLSK memories are copied into images. I/O requests addressed to output devices, such as printers (DRUA), visual display units (DSSK), are executed by emulation procedures.

Performance

Emulation generally affects performance, but as today's processors and I/O equipment offer much more performance than the components of the SICOMP system, emulation based on state-of-the-art computer technology will actually lead to a substantial increase in plant performance.
If a large SICOMP system with a high number of devices is replaced, the computers running the emulation should be capable of delivering the required performance. An appropriate hardware configuration would therefore comprise the following:

- 2 processors (dual core) with x GHz (state-of-the-art technology)
- 1024 MByte main memory
- Fast hard disks with sufficient storage space

It has been shown that plants based on current computer technology offer a substantial increase in performance - often a multiple of the performance that a SICOMP M80 system can deliver.

With today's fast hard disks (RAID systems) and the efficient disk access optimization of the Windows operating system, an even further increase in performance can be achieved for applications requiring frequent disk access.

2 Functional Description

2.1 Software Structure

The M2000 emulator for SICOMP M/R comprises an emulator core process and different peripheral processes. These processes are installed under Windows and run concurrently with other Windows processes according to their priority.

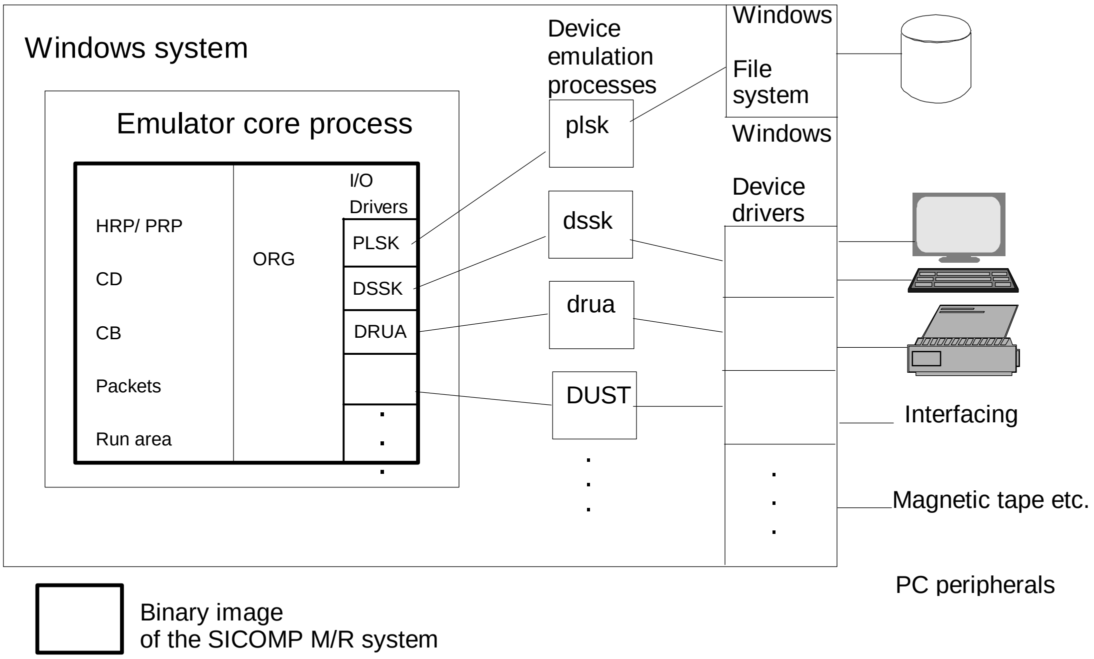


Figure 2: Software structure

The emulator core process processes the SICOMP M/R commands, comparable to the microcode of a SICOMP M/R processor, and simulates the hardware priority levels and the interrupt control of a SICOMP M/R computer. This way, processes can be managed in the same way as on the original SICOMP M/R computer. The communication between the 'ZE' (emulator core process) and the peripheral processes is handled via the same process blocks of the SICOMP system.
The device emulation processes map the SICOMP devices on the devices of the Windows system, e.g., a sub-device of disk storage is mapped in a file in the Windows file system. Thus, by applying logic mapping, the type of the physical device is of no relevance.

The emulation consequently simulates the hardware characteristics of a SICOMP M/R central unit. It is thus irrelevant which software or operating system the user system to be emulated employs. All that is required is a code that runs on the SICOMP M/R and that the used peripheral devices are supported by an M2000 peripheral process.
(see section: Scope of Performance)

The emulator core process also emulates the virtual console.

2.2 Scope of Performance

2.2.1 CPU Emulation

SIC-M/R

M2000 emulates the command set of the SICOMP M/R central units (ZE) and supports the services of the individual units. The central units to be emulated are selected individually (ZE01...ZE03, R10, R10V, R20, R30) so that always the command set and range of services of the selected unit is made available.
The floating-point arithmetic, too, is emulated for each central unit individually. M2000 does not adopt the floating-point arithmetic of the Intel processor - it simulates the floating-point of the SICOMP M/R computer to make sure that the accuracy of the specific central unit type is exactly implemented.

2.2.2 Device Emulation

The interface between CPU emulation and device emulation is the I/O job, which is interpreted and processed by the device emulations. The device emulations emulate a certain part of the firmware of the SICOMP M/R device interfacing.

SIC-M

I/O job processing is based on the SICOMP M communication interface (KOSS).
Upon processing of the SICOMP M command EAS, the CPU emulation activates the device emulation.

Video terminals

Video terminals are always connected by means of serial connection using the serial ports of the Windows system COM1: to COM9: or \\.\COM10..999.
The following video terminals or their emulations are supported by serial interface connection:

- ZBE3974R, ZBE3974MT
- DS075, DS075F
- DS075-DISIT
- DS078, DS081
- VDU2000

DS075 terminals can also be connected to a serial interface on a remote Windows computer communicating over a LAN network.
(see section Interface Concentrator).

Video terminal emulations

IPKS provides video terminal emulation in conjunction with M2000. The following video terminal emulation is available:

- Alpha Terminal Emulation TE2000 AX
  TE2000 AX emulates ZBE3974R or DS075 video terminals
- Alpha Terminal Emulation TE2000 FG (with extended scope of functions)
  TE2000 FG emulates DS075 FG video terminals
- DISIT Terminal Emulation TE2000 DX
  TE2000 DX emulates ZBE3974MT or DS075 DISIT video terminals
- Terminal Emulation TE2078
  TE2078 emulates DS078 video terminals

The video terminal emulations can be operated both in window mode and in full-screen mode.

In conjunction with M2000 the video terminal emulations can be used as follows:

- Locally on the same system as M2000 on a Windows platform
- On a separate computer by means of serial connection
- On a separate computer by means of a LAN connection

Note

You find a detailed description in the manuals:
TE2000/TE2078 Terminal Emulation

Console emulation

The Windows console, i.e. the PC screen and keyboard, can be used as virtual console, standard signaling or standard operator device.
M2000 provides a DS074 device emulation for this purpose. The device emulation can be operated both in window mode and in full-screen mode. In window mode, this terminal emulation can be activated several times. The console emulation contains a subset of the functions of the TE2000 AX Alpha Terminal. If the console is used for the operation of more demanding SICOMP application systems using functions such as definable/loadable function keys, the terminal emulation TE2000 AX or TE2000 FG should be employed.

Hard disks

The PLSK device emulation maps a sub-device of a SICOMP M/R hard disk into a Windows file. This file then represents a SICOMP M/R disk.
After the disk is copied to the file using a magnetic tape cassette or MO disk, the file contains the same data as the M/R disk, including the disk journaling system.
The interface is the I/O job. This way it is ensures that the peripheral memory management does not have to be emulated but runs as part of the ported SICOMP M/R system under the CPU emulation.
The difference to an original disk drive is usually not noticed by the operator.

- Data access to the disk is executed in the same sequence as for the SICOMP M/R.
- The use of the parameter **mode=share** in the configuration file *mpar.sys* / *rpar.sys* allows you to access the disks from both system environments (SICOMP and Windows). If this parameter is not set, the occupation mode *exclusive* will be active (default setting).
  Mode =*share* must be used if the disks of an emulated SICOMP system are accessed via the Windows program *sicview.exe* (see section Auxiliary Programs). This mode can also be used to save disks of an emulated SICOMP system under Windows without having to terminate the emulator. In this case make sure that all DVS files (for example) are closed, or that the user system is not started.
- The contents of the data medium is continued to be managed by the SICOMP M/R file management or DVS.
- DVS data media will also be processed.
- The PLSK emulation uses the Windows file system. Due to intelligent disk caching, fast disks and controllers and the SCSI bus, access and transfer rates are much higher than on the original system. The use of RAID systems substantially increases data integrity.

It is not possible to connect SICOMP-M/R drives directly to the PC with emulator.

Printers

All printers with Centronics interface can be connected in line with application requirements. The following printers can also be connected:

- DR202 printer
- 3915 Tally
- 3916 drum printer
- 3918 UD3
- 3919 PT80i

DR202 printers can also communicate via the serial interface.
The serial interface is located on the local M2000 computer or on a remote Windows computer connected via LAN.
(see section: Interface concentrator).

Note

Only printers operated in ASCII mode by default and with their own character set should be employed.
M2000 allows you to send hardcopy outputs to a connected EPSON compatible printer (the EPSON printer must be specified in FACO30).
Control characters to the printer (e.g., for typeface switching) are not modified by M2000. It may be necessary to adapt the application programs if it is not possible to use the same or a compatible printer in connection with the M2000 emulator.

Timer

The ZIG device emulation maps I/O jobs to the timer on to corresponding Windows call instructions. Due to the use of standard hardware and the Windows operating system, limitations with regard to time of day granularity and shortest possible cyclic time interrupt must be taken into account. A resolution of 10ms (or 1ms for ZIG3) represents the shortest time unit possible.

Magnetic tape cassette drive

The operation of the computer's magnetic tape cassette drive is based on an MK82 emulation. It is possible to read and write magnetic tape cassettes of 150 MB capacity. Data exchange with a SICOMP system is thus possible.

The compatible magnetic tape cassette drives specified offer the following characteristics:

- Physical Read and Write
- Logic Read and Write
- Read and Write of DVS volumes

Note

Magnetic tape cassettes generated with MK80/81 drives can be read by the MK82 emulation of M2000. However, the magnetic cassette drives on the PC are not able to write the Mk81 format.

Floppy disk

M2000 operates the 3.5 inch (1.44MB) or 5.2 inch (1.2MB) floppy disk drive as FD044 with the SICOMP M/R initialization. Data exchange with a SICOMP M/R system is however not possible.

Computer interfacing (M)

DU02

The DU02 gateway functionality (fiber optics interface connection) is implemented. It is thus possible to connect two emulator systems.

DU03

M2000 allows the emulation of a DU03 computer interface connection via a specific driver (NDIS/NDIS2000). A packet or NDIS driver must be installed on the emulator computer, which is presently available for the operating systems WindowsNT and Windows2000. The DU03 emulation completely maps the original -Data Transfer- functionality. Test and maintenance jobs are not mapped but will be processed and completed without indication for compatibility reasons.
The emulation of the DU03 requires no special hardware - all that is required is a common network card (an emulated DU03 can thus be operated with 100 Mbit/s). The driver enables the DU03 emulation to read and check the relevance of the received data packets via the corresponding network card, and to transmit data packets.

DU04

M2000 allows the emulation of a DU04 computer interface connection via a serial interface. The following three emulation methods are available:

- Emulation via any serial interface (COMxx)
- Emulation via a DF32/42 interface module
- Emulation via TCP/IP in a LAN network or via RAS.

The DU04 device emulation can be set up for TCP/IP communication to allow two M2000 systems to communicate with each other for example, or to implement data exchange with third party systems (UNIX).

A DU04 is generated in the AMBOSS/BS-M system for that purpose. The device emulation then converts this communication to the Windows socket interface (Winsocket). A LAN/RAS network adapter according to the Windows hardware compatibility list is used instead of the COMxx interface or the DF32/42.

DU05

Emulation of a DU05 computer interface connection. The DU05 emulation processes the MSV2 transmission protocol.

The following interface connections can be implemented:

- M2000 - M2000 via MSV2 and SINEC
- M2000 to third-party systems via MSV2
- M2000 via TCP/IP in a LAN network or via RAS.

DU05 emulation requires a DF42 data transmission module with the appropriate protocol driver.

The DU05 device emulation can be set up for TCP/IP communication to establish the communication between two M2000 systems, for example. A DU05 is generated in the AMBOSS/BS-M system for that purpose. The device emulation then converts this communication to the Windows socket interface (Winsocket). A LAN/RAS network adapter according to the Windows hardware compatibility list is used instead of the DF42 module.

DU06

Emulation of a DU06 computer interface connection. The DU06 emulation processes the transmission protocols HDLC U/P, HDLC U/S and HDLC B.

The following interface connections can be implemented:

- M2000 - M2000 via HDLC B, HDLC U and SINEC
- M2000 - IBM via SDLC and SINEC SNSNA
- M2000 - X25 Net via HDLC B and SINEC SNPV
- M2000 via TCP/IP in a LAN network or via RAS.

The WKO function are not implemented (Remote loading...).

The DU06 emulation requires a DF32 or DF42 data transmission module DF42with the appropriate protocol driver.

The DU06 device emulation can be set up for TCP/IP communication to establish the communication between two M2000 systems, for example. A DU06 B is generated in the AMBOSS/BS-M system for that purpose. The device emulation then converts this communication to the Windows socket interface (Winsocket). A LAN/RAS network adapter according to the Windows hardware compatibility list is used instead of the DF32/42.

PARIF interfacing
(replacement)

3961 gateway function to PARIF(**PAR**allel**I**nter**F**ace).
This interface connection establishes a transmit and receive connection to a gateway computer that communicates with an original SICOMP R computer via a PARIF module. In the emulated ORG, this interface connection is generated as DUST 3961 and operated as KNWE/KNWA.

PE3600

The process unit is emulated from the SIMATIC S7 spectrum via Profibus peripherals. This way, the PE3600 can be replaced by modern and cost-effective standard technology without having to change the SICOMP R software.

The Profibus is connected via a ProfiBus communication processor based on the appropriate software from the SIMATIC NET spectrum.

The Profibus I/O modules are operated as distributed peripherals in the ET200 system.

The following process signal converters are presently supported:

• Digital input	3611, 3612, 3613, 3615	
• Digital output	3621, 3622, 3625	
• Analog input	3631	
• Analog output	3652	
• Test detector	3668	(simulation)

PE F7

The PE F7 process unit of a SICOMP M is emulated via the Profibus.
M2000 provides a **PE F7** device emulation for I/O access to the process peripherals. This device emulation directly processes the process calls that are not allocated to a process device generated in ORG-M. Process calls using the process device ALEM are processed by an M2000 **ALEM** device function. See also: Chapter 8, PE F7 Emulation

The following process signal converters are presently supported:

• Digital input	430	
• Digital output	451	
• Digital input/output	DEDA 482	
• Analog input	460	
• Analog output	470	
• Counter module	IP 242 A	(limited functionality)

PROMEA1

Devices connected in the original system by means of a PROMEA1 module are connected to the COM interfaces of the PC and operated in the emulator via the device processes for printers, video terminals and DUST3964R.

PROMEA MX

M2000 provides the emulation of a definable PROMEA module.
The PROMEA module is operated via a serial standard interface (COMxx). The serial interface is located on the local M2000 computer or on a remote Windows computer connected in a LAN network. (see section Interface Concentrator).

Computer interfacing (R)

DUST3961

The DUST 3961 gateway functionality can be used to connect two emulator systems.

DUST3962

The DUST 3962 gateway functionality (fiber optics interface connection) can be used to connect two emulator systems.

DUST3964

M2000 allows the emulation of a DUST3964 computer interface connection via a serial interface.

DUST3964R

M2000 allows the emulation of a DUST3964R computer interface connection via a serial interface. The following two emulation methods are available:

- Emulation via any serial interface (COMxx)
- Emulation via TCP/IP in a LAN network or via RAS.

The DUST3964R device emulation can be set up for TCP/IP communication to establish the communication between two M2000 systems, for example.
A DUST3964R is generated in ORG for that purpose. The device emulation then converts this communication to the Windows socket interface (Winsocket). A LAN/RAS network adapter according to the Windows hardware compatibility list is used instead of the COMxx interface.

DUST3965R

Emulation of a DUST3965R computer interface connection. The DUST3965R emulation processes the MSV2 transmission protocol.

The following interface connections can be implemented:

- M2000 - M2000 via MSV2 and SINEC
- M2000 to third-party systems via MSV2
- M2000 via TCP/IP in a LAN network or via RAS

The DUST3965R emulation requires the DF42 data transmission module with the appropriate protocol driver.

The DUST3965R device emulation can be set up for TCP/IP communication to establish the communication between two M2000 systems, for example.

A DUST3965R is generated in ORG for that purpose. The device emulation then converts this communication to the Windows socket interface (Winsocket). A LAN/RAS network adapter according to the Windows hardware compatibility list is used instead of the DF42.

DUST3966

Emulation of a DUST3966 computer interface connection. The DUST3966 emulation processes the transmission protocols HDLC U/P, HDLC U/S and HDLC B.

The following interface connection can be implemented:

- M2000 - M2000 via HDLC B, HDLC U and SINEC
- M2000 - IBM via SDLC and SINEC SNSNA
- M2000 - X25 Net via HDLC B and SINEC SNPV
- M2000 via TCP/IP in a LAN network or via RAS.

The WIKOP functions are not implemented (Remote loading...).

The DUST3966 emulation requires the DF32 or DF42 data transmission module with the appropriate protocol driver.

The DUST3966 device emulation can be set up for TCP/IP communication to establish the communication between two M2000 systems, for example.
A DUST3966 is generated in ORG for that purpose. The device emulation then converts this communication to the Windows socket interface (Winsocket). A network adapter according to the Windows hardware compatibility list is used instead of the DF32/42.

KS100

Emulation of a SINEC H1 interface connection based on the ISO protocols up to and including the level 4.
This device emulation requires the installation of the following communication processor on the Windows platform:

- CP1413/CP1613 with the SIMATIC NET software package

CP1400

Emulation of a SINEC H1 interface connection based on the ISO protocols up to and including the level 7. Levels 5 to 7 correspond to the SINEC AP 1.0 automation protocol.

CS275

Connection of an emulated SICOMP system to the TELEPERM M bus system CS275. The implementation of the M2000 device process requires a special Siemens module (NAT).

ETC M

PromeaNET (*)
The PromeaNET function is similar to that of the SINEC ETC M for visual display units and printers. The ETC M compatible terminal emulations DEnet75(*) for DISIT terminals and DSnet75(*) for DS075 video terminal are used on computers based on MS-DOS and Windows 3.1.

Together with SPOOL75(*), such a terminal can also be used as printer server for serial printers from the SICOMP range, such as the DR216N.

Benefits and special features:

- Each generated DSSK and DRUA channel can be allocated to the PromeaNET without having to regenerate the ORG.
- 129 channels can be theoretically arranged.
- An emulated terminal can operate several emulations, even combined with SICOMP M, if these are equipped with an ETC M.

(*) Product of the SIG Aachen company

The Promea NET function is a product of the SIG Aachen company and can therefore not benefit from the M2000 guarantee and support services. If support is required, please contact SIG Aachen company directly. Product faults or operational problems will be passed on by IPKS.

Dongle

IPKS supplies different types of dongles:

- NormalDongle (options on customer request) for continuous operation, available in two versions: 25-pole plug connector/socket, USB connector
- EmergencyDongle = NormalDongle for time-limited operation of the emulation (30 days max.)
- EvaluationDongle (options on customer request) for emulation testing

Emergency dongle

The emergency dongle contains a validity period of 30 days max. If the emulator detects an unused emergency dongle upon start up, the message '*Inactive E-dongle*' is displayed. If the emulator is then closed within 30 minutes, the emergency dongle remains unused.
If the emulator runs longer than 30 minutes, the dongle becomes 'active'. The emulator then calculates the end of the validity period and displays the last valid day in a message: '*E-dongle valid until DD/MM/YYYY'*.
If the emulator detects an active emergency dongle upon start up that is still valid, the message '*E-dongle until DD/MM/YYYY'* is displayed.
If the emulator detects an active emergency dongle upon start up that is no longer valid, the message '*E-dongle expired'* is displayed*.*
At the last day of the validity period, the message '*E-dongle expires today!!!'* is displayed after every hour.

The emulator is <u>not</u> terminated should the validity period of the emergency dongle be exceeded during emulator operation.

Evaluation dongle

The evaluation dongle registers the emulator after an uninterrupted operation of one hour, and terminates the emulation without warning after another hour has expired.
An evaluation dongle can be used as often as required, but always only for two hours at the time.
In an evaluation dongle too, the required options must be enabled.

Windows

The system name Windows denotes the Microsoft operating systems Windows2000 / 2003Server and Windows XP.

3 Commissioning

3.1 Scope of Delivery

Scope of delivery (M2000)

The M2000 package includes the following:

- CompactDisc containing the M2000/PCSIC-M/R software
- One dongle
- Manual on CD

Files

The **CD** contains the following files:

readme_M2.wri	README file informing about new developments, tips and tricks
M2000.pdf	M2000 documentation
Eula_D.rtf	EndUser license in German
Eula_US.rtf	EndUser license in English
Version.txt	Information about the versions on CD

Directory *Utils\*	*mcsave.exe*	Magnetic tape (MK82) program
	pc_rdisk.exe	Copy SICOMP disks from MO disk
	xdong.exe	Issue dongle data
	diagnostix.exe	Aladdin System Data Collection

Directory *Emulation\Doc\*		**M2000 documentation**
Directory *Remote_Dongle\*		**Drivers Network dongle + documentation**
Directory *Source\*		**Replacement parts**
	M2000\binm	Not packed
	M2000\binr	Not packed
	M2000\Desktop	Not packed
	utils\	Not packed

Directory *M2000\*		The **M2000 installation** generates the following directories:
	Desktop	**M2000 programs**
	M2000.exe	Central application
	M2000hsp.exe	M2000 HSP viewer
	M2000off.exe	Emergency stop Start Process
	MPseudo.exe	DialogProgram MPseudo
	M2000.dll	Library interface M2 interface emulation
	M2000Int.dll	Library M2 interface
	M2000hsp.dll	Library HSP viewer
	readme_m.wri	Latest info on SIC-M
	readme_r.wri	Latest info on SIC-R
	readme_M2.wri	Latest info on M2000
	Help	M2000 online help
	Sicview	SICOMP hard disk viewer
	TrcView	Program TraceView
	Wmf_ico	Internal M2000 icons
Doc\		**M2000 documentation**
Binm\		**Modules SIC-M emulation**
Binr\		**Modules SIC-R emulation**
Samples\		**C++ examples for PSD/PSxKOMM**
	Inc	Include files
	Lib	Libraries
	Psexampl	Examples ORG/DVS calls via PSAPI
	Psxkomm	Examples/structograms PipeTraffic without PSAPI
Utils\		**Auxiliary programs, DLLs** such as *sickoor, mimdrv, NDIS2000, mcsave, daytimed, ...*

The user installation generates the following directories:

Sicomp\		**Parameter files/batches , assembler examples for PSD** *du0x.par, emu_strt.bat, mpar.sys/rpar.sys, plsk.psd, setall.bat, spox.par*
	Gsb	*bench, korxxx, koryyy, pccopy*
	Kor	*korxxx, koryyy*
	Psd	Examples for PSD (sources)
	Pso	Examples for PSD (ObjCode)
	Trc	*dsskx.tko, .trc, .par* files

The files *mpar.sys / rpar.sys* and *dsskx.tko* can be adapted to the relevant system. Please make sure that these files are not overwritten when you update your system (you can rename the files with system-specific names).

Example mpar.sys → audi.sys
dsskx.tko → dssk2.tko

Note The file *plsk.psd* is delivered as SICOMP "disk", i.e., if elements or programs are to be loaded or transmitted from the "disk" libraries, *plsk.psd* must be entered as a logic disk drive in *mpar.sys /rpar.sys*.

Scope of delivery (SIC-M) The M2000 installation generates the following directory for SIC-M:

Directory ***Binm***		**ZE (central unit) / device emulations)**
	sic_m.exe	Initialization process
	sic_ze.exe	ZE emulation
	sic_zeit.exe	Device emulation Timer
	sic_disk.exe	Device emulation Disks
	sic_op31.exe	Device emulation MO disk (OP31)
	sic_eads.exe	Device emulation Serial interfaces
	sic_eadr.exe	Device emulation Printers
	sic_eamx.exe	Device emulation Promea MX
	sic_eaes.exe	Device emulation ES terminals
	sic_tape.exe	Device emulation Magnetic tape cassette
	sic_flop.exe	Device emulation Floppy disk drive
	sic_du03.exe	Device emulation DU03
	sic_du04.exe	Device emulation DU04
	sic_du05.exe	Device emulation DU05
	sic_du06.exe	Device emulation DU06
	sic_du63.exe	Device emulation 3963/BiSBox
	sic_du78.exe	Device emulation DS078
	sic_rb06.exe	Remote boot function
	sic_rs06.exe	Remote boot function
	sic_ucp2.exe	Device emulation TCP/IP via UCP2
	sic_ftnt.exe	Device emulation Pseudo device
	sic_ks10.exe	Device emulation KS100
	sic_ks11.exe	Device emulation KS100
	sic_disi.exe	DISIT terminal connection
	sic_coro.exe	VD2000 connection
	sic_etcm.exe	Connection SIG terminal emulation via ETC M
	sic_cp14.exe	SINEC H1 interface connection via ISO
	sic_cs27.exe	Connection to TELEPERM M
	sic_spox.exe	Printer output to Windows file

Scope of delivery (SIC-R) The M2000 installation generates the following directory for SIC-R:

Directory ***Binr*** **ZE (central unit) / device emulations, Psf3600.dll**

sic_r.exe	Initialization process
sir_ze.exe	ZE emulation
sir_zeit.exe	Device emulation Timer
sir_disk.exe	Device emulation Disks
sir_eads.exe	Device emulation Serial interfaces
sir_eadr.exe	Device emulation Printers
sir_eaes.exe	Device emulation ES terminals
sir_eamx.exe	Device emulation Promea MX
sir_flop.exe	Device emulation Floppy disk drive
sir_3962.exe	Device emulation DUST3962
sir_3964.exe	Device emulation DUST3964R
sir_39640.exe	Device emulation DUST3964
sir_3965.exe	Device emulation DUST3965
sir_3966.exe	Device emulation DUST3966
sir_61gw.exe	Device emulation 3961 gateway
sir_cs27.exe	Connection to TELEPERM
sir_rb66.exe	Remote boot function
sir_rs66.exe	Remote boot function
SIR_FTN.exe	Device emulation Pseudo device
sir_ftnt.exe	Device emulation Pseudo device
sir_wtf.exe	Maintenance panel
sir_xpsa.exe	Device emulation PSA
Psf3600.dll	

3.2 System Requirements

Conversion with M2000 requires that the originating system is a Siemens SICOMP M/R system with MKSK. All system and data disks of the SICOMP M/R will be transferred to magnetic tape and then stored in Windows files on the PC.
The PC configuration consists of the following hardware components:

Processor

The PC must be equipped with an Intel i486 or Pentium processor (or compatible processor). Although Windows can be operated on hardware platforms that are not based on Intel processor architecture, M2000 can only be used with Intel-compatible processors. The SICOMP M/R commands are directly mapped to the Intel command set to obtain optimum performance.

Important! Make sure the internal processor cache is not deactivated as this would increase the emulator processing times by factor of five. The use of an external cache will further enhance the system's time behavior.
M2000 requires a minimum main memory space of ≥ 64 MB. It must be made sure that all M2000 components can run in the RAM memory and are not paged out to the Windows paging file.

Graphics card and keyboard

Must correspond to Windows requirements.

Hard disk

As the data are accessed via the Windows file system, sufficient hard disk space must be provided to allow all data disks to be stored in one and the same partition. Different PLSK can be stored on the same or on different Windows logic drives.

Serial interfaces

Serial interfaces corresponding to Windows requirements must be provided for the connection of the terminals.

Terminals

The following options are available for the operation of the emulated SICOMP M/R:

- Connection of video display terminals via serial interfaces
(local and remote, see Interface Concentrator), i.e. the existing video terminals can continue to be used.
- Local terminal emulations operated on the emulation system with the Windows console. The Windows console can also be used as virtual console, standard signaling or standard operator device.
- Terminal emulations via serial interface or network connection

Printers

All printers with Centronics interface can be connected in line with application requirements, also DR202 printers (including 3918, 3919) connected via serial interfaces.
(local and remote, see section: Interface Concentrator).
Only printers operated in ASCII mode by default and with their own character set should be employed..

Streamers

The MKSK emulation of M2000 support streamers of the type Tandberg Data Panther TDC 3660 or TDC 3820.

Floppy disk

3.5 inch 1.44 MB or 5.25 inch 1.2 MB
(SICOMP M/R formatting is used).

Windows

The PC must be based on a Windows 2000/2003(Server), WindowsXP platform.
An M2000 installation under WindowsXP requires the NTFS file system.

3.3 Installation

3.3.1 Overview

The individual steps for the installation of a SICOMP M/R system on an adequately equipped Windows computer (see: System Requirements) are described below.

Step 1

The provided standard or USB dongle must be plugged into the parallel printer port (LPT1) or a free USB slot, respectively, and must remain plugged for the entire validity period of the emulator. The software cannot be operated without a dongle.
If the LPT1 port is occupied by a printer, connect the printer cable to the standard dongle. Do not connect the USB dongle to a USB hub (always plug into free USB slot).

Step 2

If you use an USB dongle, install the USB dongle driver first
(**Setup – Product installation – USB dongle driver installation**).

Step 3

Install M2000 (incl. emulator) (see 3.3.2 Installing M2000)

Step 4

Install the user system (see 3.3.2.1 Installing the User System)

Step 5

Install the copy of the SICOMP M/R user system (see Online help)

The magnetic tape copies or OPxx data media generated on the originating system are read in and stored in Windows files. These Windows files then represent the data media of the SICOMP M/R system. To start the reading procedure, select `General – Copy – Import data in`.

Alternative: You can transfer the disk copies (MKxx/Opxx) of the originating system prior to the installation of M2000 with the programs *mcsave.exe* or *pc_rdisk.exe* (in the CD directory *Utils*).

Step 6

Edit the configuration file of the user system (see 3.3.3 and 4.)

M2000 requires a configuration file describing the originating system for the device configuration to be emulated.
In %M2000Disk%\sicomp\..., the example configurations *mpar.sys/rpar.sys* are provided. You can adapt these (MEDIS-compatible) files to your own system configuration. You can also create a configuration using the M2000 function `Automatic generation` on the `General` tab, and then adapt this configuration in the M2000 editor.

Step 7

Start the emulation (see 3.4)

After you completed all preparations, the SICOMP M/R emulation can now be started from the M2000 menu or `Monitor` tab.

Terminal emulation

During the installation of M2000, the current terminal emulation (*ds074.exe* or *te2000.exe*) is stored in the subdirectory .....*\Ipks\Utils\*.... You can install terminal emulations directly from this directory via a computer network. Please note that terminal emulations may only be operated in connection with the appropriate licenses (entry in the M2000 dongle or separate terminal dongle).

Note !

The functions of M2000 are described in the M2000 online help (`Help-Contents, or` F1).

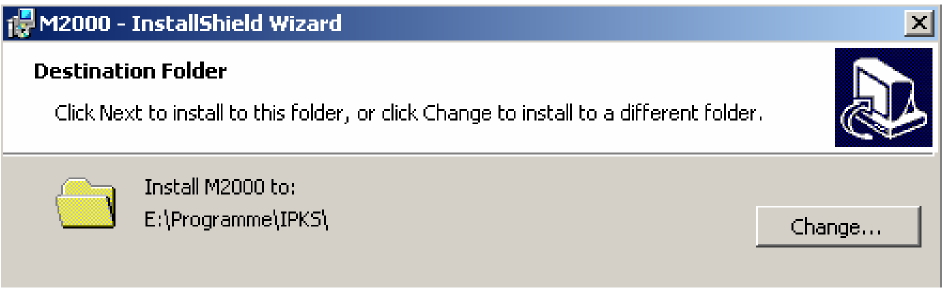
If all external USB slots are occupied, the use of an USB 2.0 PCI host adaptor is a good idea. These PCI cards provide several external USB slots and usually one internal USB slot the USB dongle can be plugged into. Internal connection has the advantage that the dongle is protected from unauthorized or accidental removal.

Important! The README file (*readme_M2.wri*) provided on the CD contains the most up-to-date information about the M2000 system. We recommend reading this information carefully.

3.3.2 Installing M2000 (incl. Emulator)

1. Log in as administrator in Windows.

2. Insert the installation CD for your M2000 software. By default (autorun), when you insert the CD, *Install Shield* will prepare your system for product installation. You can also activate *Install Shield* manually by double-clicking *setup.exe* in the *root* directory of the CD. The default installation directory is *%SystemRoot%\Program Files\IPKS\M2000\....* Click `Change` to choose a different installation directory.



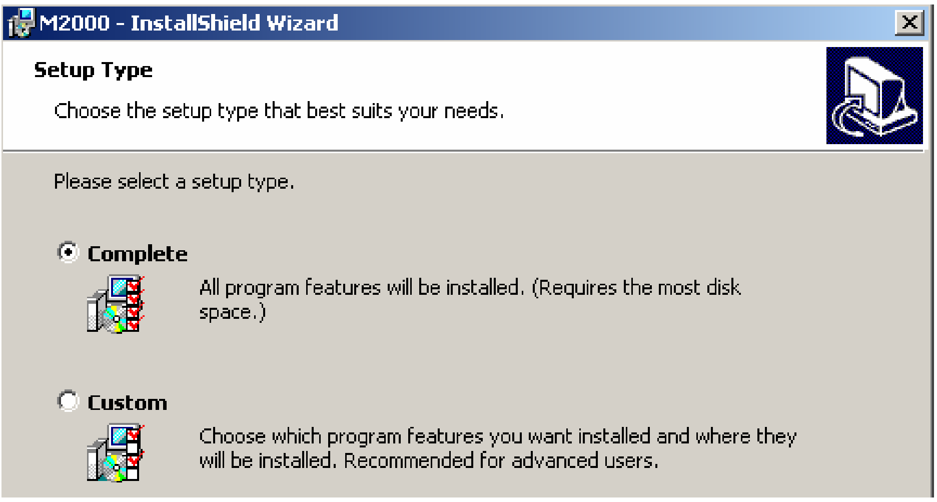
M2000 - InstallShield Wizard

Destination Folder

Click Next to install to this folder, or click Change to install to a different folder.

Install M2000 to:
E:\Programme\IPKS\

Click ● **Complete** to install all M2000 features. Click ● **Custom** to select M2000 features to install. Then follow the instructions on the screen.



M2000 - InstallShield Wizard

Setup Type

Choose the setup type that best suits your needs.

Please select a setup type.

● **Complete**
All program features will be installed. (Requires the most disk space.)

○ **Custom**
Choose which program features you want installed and where they will be installed. Recommended for advanced users.

3. Update installation

   Insert the update CD and start the M2000 setup routine. The update procedure is completed automatically. You can cancel the update procedure at any time, selection options (`change directory, select components, etc.`) will not be available.

   You can also update the system by selecting the Windows system function Start-Control Panel-Add/Remove Programs-Change/Remove.

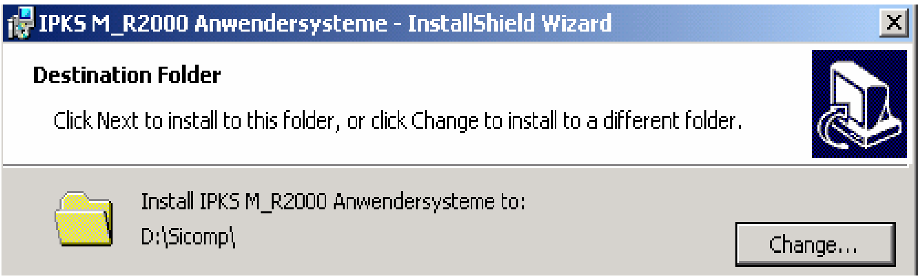
   To be able to execute the functions
   ⦿ Change program
   ⦿ Repair program
   the original installation medium must be available. Follow the instructions on the screen.

4. After a reset of the PC (Start-Shutdown-Restart), you can activate M2000 and continue with the preparations. (see 3.3.1, steps 3 and 4)

3.3.2.1 Installing the User System

1. You are still logged in as administrator in Windows.

2. To install the data of the user system on the disk, click User system in the M2000 product installation. Follow the instructions on the screen. If you selected the standard option for the M2000 installation (● **Complete**), the data are automatically installed to the location *%M2000disk%\Sicompl\*. Click **Change...** if you want to choose a different location. The **Change...** button for choosing a different destination is also available for the user-specific installation (● **Custom**).

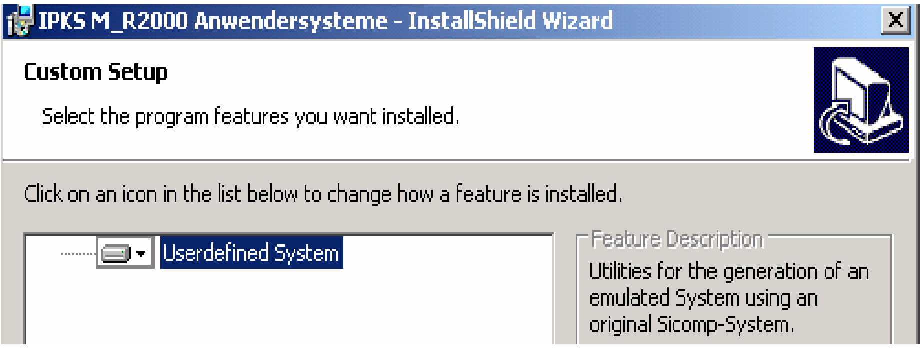


IPKS M_R2000 Anwendersysteme - InstallShield Wizard

Destination Folder

Click Next to install to this folder, or click Change to install to a different folder.

Install IPKS M_R2000 Anwendersysteme to:
D:\Sicomp\



IPKS M_R2000 Anwendersysteme - InstallShield Wizard

Custom Setup

Select the program features you want installed.

Click on an icon in the list below to change how a feature is installed.

Feature Description

Utilities for the generation of an emulated System using an original Sicomp-System.

3. The user system, i.e. parameter files such as *mpar.sys /rpar.sys*, *emu_strt.bat*, auxiliary programs in *plsk.psd* etc., will be stored in %M2000dir% - (e.g.,. C:\Sicomp\). Copy the SICOMP data media to %M2000dir%\disks\see:

   3.3.1 Overview, step 4

3.3.2.2 Changed System Options

After the installation of M2000, the following system options have changed:

- - Control panel – System – Advanced - Performance -
  `Optimize system performance for background services`
  (Default: `.... for applications)`
- – Event indication – Application log/ system log – Properties -
  If maximum log size is reached:
  overwrite events as needed
  (Default: overwrite events older than `...)`

- - Display properties – Appearance - Effects -
  ☐ `Show window contents while dragging` - is deactivated.
  (Default: ☑ `Show window contents while dragging)`

- AutoUpdate function in the Registry of Windows2000SP3/WindowsXP is deactivated.
  (HKEY LocalMachine\System\CurrentControlSet\Services\wuauserv ▸ Start:)

3.3.3 Configuration File (mpar.sys/rpar.sys)

The configuration file contains information about the computer to be emulated and the assignments of the devices used. Examples files are provided that can be used as templates for your individual configuration. In the `Configuration` section of the M2000 `Monitor` tab, click `New` or `Change` to allocate or edit a template configuration file. Refer to the Online help of M2000 for additional information.

The configuration file consist of individual lines (text file). Each entry in the configuration file occupies a separate line with a maximum length of 250 characters. The provided example files have a line width of 80 characters and do not contain vowel mutation (umlaut). The files can also be edited with the MEDIS editor.

Every line (parameter section) should end with a semicolon. The semicolon denotes the beginning of the comment section (this line section will be overlooked by the emulator). Blank lines are permitted. No distinction is made between small and large caps. Every entry starts with a code word followed by one or more parameters. The entries in the configuration file should be arranged as follows:

- Sorting by device number in ascending order by ...(*ioadr-devicenr* or *anr,gnr*)
- For disk drives: *ioaddr-devicenr* <u>must</u> be edited in ascending order
- End in a semicolon (parameter section)
- Add comment

Syntax errors or logic errors in the lines of the configuration file will be documented in the event log.

The exact notation of the entries in the configuration file is described in the chapters 4 and 5.

General entries:

CPU	CPU type	(M/R)
ZESTART	ZE (central unit) start delay	(M)
MAXMEM	Max. main memory	(M/R)
IMAGE	Main memory retrieval file	(M/R)
BOOT	Address bootstrap loader and virtual console	(M/R)
MECR	Operation of an MEC-28R interface connection	(R)
NOFPP	Deactivate floating-point processor	(R)

Device definition entries

Device definition entries are used to allocate a SICOMP M/R device to a device of the Windows system.
In the configuration file, a DEVICE entry must exist for every SICOMP M/R device. The DEVICE entries should have the following structure:

SIC-M

DEVICE = *ioaddr[-devnum],keyword,"pathname"*
DEVICE = *ioaddr[:m],keyword,"pathname"*
DEVICE = *ioaddr[:m],keyword,"pipename"*
DEVICE = *ioaddr[:m],keyword,"socketnr"*

ioadr is the I/O address of the device according to ORG generation (4-digit hexadecimal number). For devices for which a *Device Number* can be defined (such as disk drives), the optional parameter *devnum* can be specified. If not all of the 8 channels/sub-devices (I/O block operation, character operation, virtual console, etc.) of an I/O address are used for devices such printers, video terminals, Promea MX, Promea ES, FTNT, then more I/O addresses can be generated in ORG. The parameter *m* is then used to specify the distance (generated earlier in ORG) between two I/O addresses. Possible values for *m*:
0 – One-step division of the *ioaddr* `device=A380:0,... =A381:0,...`
1 – Two-step division of the *ioaddr* `device=A380:1,... =A382:1,...`
3 – Four-step division of the *ioaddr* `device=A380:3,... =A384:3,...`

Examples

DEVICE = 8200**-7**, FP023, "\sicomp\disks\plsk007"; --- -devnum
DEVICE = A378**:1**, DRSPOTS, "I:\printers\Nsu",30; --- Two-step
DEVICE = A37A**:1**,DRSPOTS, "I:\printers\In",30; --- Two-step
DEVICE = A100**:0**,DR202, "..\\.\COM182"; --- One-step
DEVICE = A101**:0**,DR202, "..\\.\COM183"; --- One-step
DEVICE = 8008,DS074, **"DSSK0"**; --- Pipe name
DEVICE = A390:1,DS074NET, **"20002"**; --- Socket no.

The parameter *keyword* designates the device in Windows. *Pathname* allocates a Windows device, *pipe name* is the designation of the named pipe for a local terminal emulation, and *socketnr* the socket number for a terminal emulation via network.

SIC-R

DEVICE = *anr,gnr,keyword,"pathname"*
DEVICE = *anr,gnr,keyword,"pipename"*
DEVICE = *anr,gnr,keyword,"socketnr"*

anr The connecting point number of the device according to ORG generation (decimal number).
gnr The device number according to ORG generation (decimal number).

keyword The parameter designates the device in Windows.

pathname The parameter allocates a Windows device.

pipename Unambiguous designation of the *named pipe* for local terminal emulation

socketnr Socket number for a terminal emulation via network

Examples

DEVICE = 5,2, 3948, "\sicomp\disks\Rsys0.v1";
DEVICE = 1,1, DRCOM, "COM7",3915,,100
DEVICE = 1,2, DR202, "\\.\COM17";
DEVICE = 2,0, 3974RT, "DSSE1";
DEVICE = 2,1, 3974RNET, "20001";

Definable devices

<u>SIC-M</u>

3974MT Operation of a ZBE3974MT or DE75, or a compatible emulation, via serial interface.
Operation of a TE2000 DX DISIT emulation as local terminal.

3974MTNET Operation of a TE2000 DX DISIT emulation as network terminal.

DS074 Operation of a console emulation (local terminal, VK).

DS075[F]LOK Operation of a TE2000 AX Alpha terminal emulation (color) as local terminal.

DS075[F]NET Operation of a TE2000 AX Alpha terminal emulation (color) as network terminal.

DS075NET Operation of a TE2000 FG Alpha terminal emulation as network terminal.
DS075 Operation of a TE2000 FG Alpha terminal emulation.

DS075[F] Operation of a DS075 (color) via serial interface.

DS075 DISIT Operation of a DS075 DISIT via serial interface. (TE2000 DX)

DU78 TE2078 emulation of a DS078 terminal.

VDU2000 Operation of a VDU2000 via serial interface.

PS048 Emulation of a sub-device of a PS048(32 MB) disk drive by means of a Windows file.

PS049 Emulation of a sub-device of a PS049(13 MB) disk drive by means of Windows file.

FP0XX Emulation of an FP023, FP024, etc. (hard)disk drive partition by means of a Windows file (variable length).

FP0XXAE Emulation of an FP023, FP024, etc. (hard)disk drive partition with ASCII-EBCDIC conversion by means of a Windows file (variable length).

FLOP 3½ inch floppy disk drive, <u>not</u> compatible with SICOMP M.

OP31 Operation of an OP31 optical disk. (M2-MSE and higher)

ZIG1 Emulation of a Promea timer.
ZIG2 Emulation of a Promea timer (ms accuracy for time reading only).
ZIG3 Emulation of a Promea timer with millisecond accuracy.

DRUA Conversion of a printer to the parallel interface or a Windows file.
DR202 Operation of a DR202 printer via serial interface.
DRCOM Operation of all (serial) printers via serial interface.
DRSPOOL Printer output via a Windows printer driver.
DRSPOOLF Similar to DRSPOOL, but with FF(FormFeed) at the output end (laser printer).
DRSPOT Printer output to a Windows file (not DOS compatible).
DRSPOTS Printer output to a Windows file (DOS compatible).
DRSPOTX Printer output to a Windows file (with parameter file).
DRWIN Printer output to a Windows window.
DRWINT Printer output to a Windows window with date/time specification for every line.

MK82 Operation of an MK82 compatible Tandberg Data Panther 250SE magnetic tape cassette drive.

EAMX Emulation of a Promea MX.

EAES Connection of ES200 terminals.

DU02 Emulation of a DU02 computer link interface. (M2-LAN)

DU03 Emulation of a DU03 computer link interface. (M2-LAN)

DU04 Emulation of a DU04 computer link interface. (M2-WAN)

DU05 Emulation of a DU05 computer link interface. (M2-WAN)

DU06 Emulation of a DU06 computer link interface. (M2-WAN)

ALEM Emulation of a PE F7 processor via ProfiBus. (M2-PEPB)

KS100 Emulation of a KS100 communication control . (M2-LAN)

CS275 Connection to TELEPERM M. (M2-CS)

CP1400 Emulation of a SINEC H1 interface connection (ISO 1-7).

UCP-2 Emulation of an UCP-2 communication processor (M2-LAN)

FTNT Pipe interface for the data exchange between SICOMP and Windows (see PSD manual). (M2-OPEN)
Pipe interface for the data exchange between two SICOMP systems under M2000 via a Windows network.

WINCC Communication optimization between internal timers and WINCC-PSD-DLL.

ODBC Extension of the Pipe interface for the data exchange between SICOMP M/R and Windows with an ODBC client. (M2-OPEN)

PSCU Communication channel between SCU and SICOMP programs.

Note!

DEVICE NAME (boldface) **Optional package**. Must be ordered separately and enabled in the dongle.

SIC-R

3974M Operation of a ZBE3974M or DE75, or compatible emulation, via serial interface.
3974MT Operation of a TE2000 DX DISIT emulation as local terminal.

3974MNET Operation of a TE2000 DX DISIT emulation as network terminal.

3974RT Operation of a TE2000 AX Alpha terminal emulation as local terminal.

3974RNET Operation of a TE2000 AX Alpha terminal emulation as network terminal.

3974R Operation of a 3974R/DS075 via serial interface.

DS075 DISIT Operation of a DS075 DISIT via serial interface. (TE2000 DX)

3941 Emulation of a 3941 disk drive with 9744 sectors by means of a Windows file.
3942 Emulation of 3942 disk drive with 48720 sectors by means of a Windows file.

3945 Emulation of a 3945 disk drive with 2048/4096/8192 sectors by means of a Windows file.
3946 Emulation of a 3946 disk drive with 8192 sectors by means of a Windows file.
3947 Emulation of a 3947 disk drive with 15616/1034240 sectors by means of a Windows file.
3948 Emulation of a 3948 disk drive with 64640/61408 sectors by means of a Windows file.
3949 Emulation of a 3949 disk drive with 25856 sectors by means of a Windows file.
FLOP 3½ inch floppy disk drive, not compatible with SICOMP R.

ZIG1 Emulation of a Promea timer.
ALE1 Emulation of a 3691A timer.

DRUA Conversion of a printer to the parallel interface or a Windows file.
DR202 Operation of a DR202 printer via serial interface.
DRCOM Operation of all (serial) printers via serial interface.
DRSPOOL Printer output via a Windows printer driver.
DRSPOOLF Similar to DRSPOOL, but with FF(FormFeed) at the output end (laser printer).
DRSPOTX Printer output to a Windows file (with parameter file).
DRWIN Printer output to a Windows window.
DRWINT Printer output to a Windows window with date/time specification for every line.

MK82 Operation of an MK82 compatible magnetic tape cassette drive Tandberg Data Panther TDC 3660, TDC 3820.
EAES Connection of BDE terminals of the 38xx family.

3961 Emulation of an interface connection via LAN (GatewayFunction). (M2-WAN)
3962 Emulation of a (fiber optics) interface connection (GatewayFunction). (M2-WAN)
3964 Emulation of a DUST3964 computer link interface. (M2-WAN)
3964R Emulation of a DUST3964R computer link interface. (M2-WAN)
3965 Emulation of a DUST3965 (MSV2) computer link interface. (M2-WAN)
3966 Emulation of a DUST3966 computer link interface. (M2-WAN)

FTNT Pipe interface for the data exchange between SICOMP R and Windows (see PSD manual).
Pipe interface for the data exchange between two SICOMP R systems under M2000 via a Windows network. (M2-OPEN)

WINCC Communication optimization between internal timers and WINCC-PSD-DLL.

EAMX1 Emulation of a Special Promea for inputs to ORG.

PE3600 Emulation of a PE3600 process unit via Profibus peripherals. (M2-PEPB)

ODBC Extension of the Pipe interface for the data exchange between SICOMP R and Windows with an ODBC client. (M2-OPEN)
OP11 Emulation of an OP11 optical disk. (M2-MSE and higher)

PSCU Communication channel between SCU and SICOMP-R programs.

Note!
DEVICE NAME (boldface) **Optional package**. Must be ordered separately and enabled in the dongle.

3.4 Emulator Operation with M2000

Before you start the emulation, the configuration files *mpar.sys* or *rpar.sys* and the StartBatches *emu_strt.bat* and *setall.bat* must be adapted to the user configuration.

Starting the Emulation

To start a SICOMP M/R emulation, click **Start** on the `Monitor` tab, or select **Emulation** ► **Start** from the `Monitor` menu.
See also: M2000 menu or `Monitor` tab, or M2000 online help. (Menu `Help-Contents` or F1, or right click to open context-sensitive direct help)

The following command line is executed in the emulator start batch (*emu_strt.bat*):

```
start "EMULATOR" /min sic_m ..\mpar.sys     (with minimized DOS window after start)
or
start "EMULATOR" /min sic_r ..\rpar.sys     (with minimized DOS window after start))
                      |       |
                      |       |___ Configuration file
                      |___ Emulator initialization process
or

sic_m ..\mpar.sys            (without DOS window after start)
or
sic_r ..\rpar.sys            (without DOS window after start )
  |      |
  |      |___ Configuration file
  |___ Emulator initialization process
```

All involved (devices)processes are displayed in the left box of the M2000 **Output** tab in a fixed format.
(name_ioadr Pid Options Traceflags) (M)
(name_anr,gnr Pid Options Traceflags) (R)

Upon starting the emulator, the SICOMP operating system is started. All subsequent operations are performed in ORG/BS-M/R notation.

Stopping the emulation

See: M2000 menu or `Monitor` tab in the M2000 help.
The current status of the emulation is indicated in the M2000 status bar.
(emulation inactive)
This corresponds to a RESET of the SICOMP M/R system with all consequences (e.g., open DVS files).

Stopping the central unit (ZE)

See: M2000 tab `Emulation` or M2000 help.

The current status of the emulation is indicated in the M2000 status bar.
(M-ZE stopped)

4 Configuration File for SIC-M

4.1 General Entries

CPU type

`CPU` = *`typ`*

The specification of the CPU type (*typ)* is particularly important for machine intimate programs and the accuracy of calculation results.

The following CPU types are supported: ZE01
ZE02
ZE03

Example `cpu = ze03;`

Main memory size

`MAXMEM` = *`size`*

Size of the main memory denoted as decimal number in the range from 64 to 4032 (in KW). This corresponds to the maximum size of 8 MB of the original system minus 64 KW read-only memory. The parameter can be taken from the generation log of the original system.

Example `maxmem = 4032;`

ZE start delay

`ZESTART` = *`n`* `(seconds)` 0 <= n <= 120

Virtual addressing

`mode = virtual`

The data exchange between ZE (central unit) and I/O device is always virtually addressed.
Mandatory parameter for HSP > 4096KW (external intermediate memory).

Main memory retrieval file

`IMAGE` = "*`pathname`*"*`[,mode]`*

This parameter defines a Windows file to which M2000 writes the main memory contents of the emulated SICOMP M upon normal termination of the program. (button/menu entry **Stop** in M2000).

`pathname` Complete path of the Windows file. The file is set up implicitly.

`mode` Processing mode of the image file.

NEW (default). The image file is generated anew under the same name with every termination of the program. An existing image file will be deleted.

ALTx The parameter ALTx (x=1 to 9) generates x image files which are alternately overwritten. The image files are designated *filename.im1....filename.im9*. The header of every image file contains the date/time specification and a consistency identifier.
At the start of the write process, an inconsistency identifier is entered. After the HSP contents has been completely stored, a consistency identifier with time/date specification is set. The oldest or an inconsistent file will be overwritten.

When the emulation is restarted, the main memory of the emulated SICOMP M systems is pre-allocated from this main memory retrieval file. The corresponding identifiers for battery-backed restart are transmitted to the ORG system and the DVS files can be reconstructed with DFCONS.

Example `IMAGE = "\sicomp\disks\sysrett.v1"`

PopUp window indicating internal problems

mode = popup

Explained in detail in the EventLog

Bootstrap loader and virtual console

BOOT = *hexboot,hexvc*

This parameter defines the *boot_device* and the virtual console.

hexboot IO address of the *boot_device* in the form of a 4-digit hexadecimal number. An additional sub-device identifier is integrated in the IO address as follows:

Device no.-sub-device no.	results in	boot device
8100 - 1		**8110**

hexvc IO address of the virtual console

Example

```
BOOT = 8110, 8008;
```

4.2 Device Definition Entries

4.2.1 Visual Display Units

Visual display units, serial connection

`DEVICE` = *`ioaddr,keyword,"pathname"`*[*`,upar`*[*`,identifier`*[*`,timer`*]]]

This parameter defines the connection of a visual display unit via serial interface (COMxx).

`ioaddr` IO address of the device according to ORG generation
(4-digit hexadecimal number)

`keyword` The following devices can be specified:
DS075 or DS075SER
DS075F or DS075FSER
3974MTSER
3974MTFSER
DS075_DISIT
DU78
VDU2000

The *keyword* parameter must be assigned as followed:

DS075, DS075SER	for TE2000 AX or FG or TE-ALPHA
DS075F, DS075FSER	for TE2000 AX or TE-ALPHA (color)
3974MTSER	for TE2000 DX or TE-DISIT
3974MTFSER	for TE2000 DX or TE-DISIT (color)
DS075_DISIT	for TE2000 DX or TE-DISIT (color)
DU78	for TE2078
VDU2000	for VDU2000

`pathname` Designates the serial interface defined in Windows. The *pathname* is denoted as "COMx" or "\\.\COMxx".
For DS075, the serial interface can also be on a remote computer. The *pathname* is specified in the form REM:x. Parameter x denotes the port number of the first of two TCP/IP sockets created by the device process on the M2000 computer.
(see section Interface Concentrator)

`upar` Mode 1: for DISIT terminal emulations only: overparameterization (hexa)
= 0: Overparameterization possible
= 1: Protect interface from overparameterization (default setting)
Mode 2: Device inoperable detection (hexa)
= 100: Device inoperable detection for TTY interface
(jumper between RxD and CTS required)
= 200: DSR scan prior to output to V.24 interface

Modes 1 and 2 can be linked by an OR operation.

`kennung` Enables the terminal emulation in the external/internal dongle
IPKS identification (kennung): Enables the functions of the (old) TE-ALPHA/TE-DISIT terminal emulation in the central M2000 dongle.
TE2000 / TE2000FG identification (kennung): Enables the functions of the (new) TE2000AX/DX/FG terminal emulation in the central M2000 dongle.
NONE identification (kennung): Function enabling in external dongle required.

`timer` Time in ms the inputs/outputs are delayed to accommodate slow visual display units (e.g., ES05) or user programs.

Notes

- The interface must be defined accordingly with the Windows *mode* command.
- Flow control is set to XON/XOFF=93/94h by M2000.
- Presently, overparameterization is only possible for the baud rate.
- A detailed description is given in the User Manual **TE2000 Terminal Emulation**.
- See also section 4.2.1 Visual Display Units, table 1.
- Commas must be entered in the parameter line even if no options are selected.

Note

During parameter definition via ORG (/PARAM ..), the M2000 driver process evaluates bytes 0,1,2,3,4. For TE2000AX/TE-ALPHA (DS075), M2000 also evaluates byte 13.

Important! The function enabling identifications (TE2000 or IPKS) are not interchangeable, they can only be used with the corresponding emulation (TE2000/TE-ALPHA/-DISIT).

Examples

Terminal connection to a local COM port.

```
DEVICE=A308:3,DS075,"\\.\COM25",101,IPKS,500
```

Terminal connection (TTY) with device inoperable detection, overparameterization not possible, central function enabling, time delay 500 ms.

```
DEVICE=A308:1,DS075SER,"\\.\COM25",200,NONE,0
```

Terminal connection (V.24) with device inoperable detection, overparameterization possible, external dongle, without time delay.

DS075F terminal connection to a local COM port.

```
DEVICE=8008,DS075F,"\\.\COM26",,TE2000
```

DS075 terminal connection to the COM port of a remote Windows system.

```
DEVICE=8008,DS075,"REM:22000"
DEVICE=A308:1,DS075SER,"REM:22000"
```

DS078 terminal connection to a local COM port.

```
DEVICE=8030,DU78,"\\.\COM7";   TE2078 terminal emulation
```

Terminal emulation, local

DEVICE = *ioadr*,*keyword*,"*pipename*"[,*timer*]

This parameter activates a terminal emulation as local Windows application.

ioadr IO address of the device according to ORG generation (4-digit hexadecimal number)

keyword The following type specifications are permitted:

DS074 / DS074F	for a local console/emulation (color)
DS075LOK / DS075FLOK	for a local console/emulation (color)
3974MT or 3974MTLOK	for a local DISIT console/emulation
DS075_DISITLOK	for a local DISIT console/emulation

pipename Unique name of the console application. A *named pipe* is created under this name, which must unambiguous throughout the system. This name is also displayed in the window heading.

timer Time in ms the inputs/outputs are delayed to accommodate slow visual display units (e.g. ES05) or user programs.

Notes

- If used in a window, the screen resolution should not be set below 800x600 pixels (16/256 colors)
- This parameter can be defined several times to emulate several display units. In this case, each device is displayed in a separate window.
- A powerful graphics card (PCI) should be used to enhance performance.
- Do not include special characters in the pipe name, such as (- \ , / . ?, etc.) to avoid that several terminal emulations work with the same pipe.
  Example: Instead of A-DSSK002, write A_DSSK002
- The device emulation's scope of performance is briefly described in section 6.1 Device Emulation DS074.
  A detailed description is given in the User Manual **TE2000 Terminal Emulation**.

Examples

```
DEVICE=8008,DS074,"DSSK0";
DEVICE=A308:3,DS074,"DSSK0";
DEVICE=8008,DS075_DISITLOK,"DSSK0";
```

DISIT emulation TE2000 DX
via network

DEVICE = *ioadr*,*keyword,*"*socketnr*[*/t*]"[,TE2000[,*timer*]]

This parameter defines the operation of a DISIT terminal emulation TE2000 DX via network

DISIT emulation TE-DISIT
via network

DEVICE = *ioadr*,*keyword,*"*socketnr*[*/t*]"[,IPKS[,*timer*]]

This parameter defines the operation of an (old) DISIT terminal emulation TE-DISIT via network.

ioadr IO address of the device according to ORG generation
(4-digit hexadecimal number)

keyword The following type specifications are permitted:

3974MTNET	TE2000 DX/TE-DISIT via network
3974MTNETF	TE2000 DX/TE-DISIT (color) via network
DS075_DISITNET	DS075-DISIT via network

socketnr Socket number for the communication between the M2000 system and the terminal emulation. Socket (port) number according to administration.
recommended range: >10000
The established connection between M2000 and the terminal emulation will be time-monitored. If no data are exchanged within **t** seconds (default value: 180s), M2000 terminates the connection. The monitoring period ***t*** can be parameterized with the socket number (see example).

TE2000 Function enabling of the (new) TE2000 DX terminal emulation in the central M2000 dongle.

IPKS Function enabling of the (old) TE-DISIT terminal emulation in the central M2000 dongle.

timer Time in ms the inputs/outputs are delayed to accommodate slow visual display units (e.g., ES05) and user programs.

Examples

```
DEVICE=8030,DS075_DISITNET,"20011";
DEVICE=A308:3,3974MTNET,"20011";     default time monitoring
DEVICE=A308:3,3974MTNET,"20011/3600";time monitoring 1h
```

DS078 emulation TE2078
via network

DEVICE = *ioadr*,DU78*,*"SOCKET",*port-o,host-remote:port-r*

This parameter defines the operation of a DS078 terminal emulation
TE2078 via network.

ioadr see above

port-o(wn) Port/socket number in the local computer for the communication between M2000 and the terminal emulation. The socket number should be specified by the network administrator.
Recommended range: >10 000

host-remote Host name or address in the local or remote computer
(e.g., emulation PC or 192.168.1.111).

port-r(emote) Port/socket number in the local or remote computer for the communication between M2000 and the terminal emulation. Socket number *port-r* must differ from socket number *port-o*.

Examples

```
DEVICE=8410,DU78,"SOCKET",20000,192.168.1.49:20001;
```

Note A detailed description is given in the User Manuals:

TE2000/TE2078 Terminal-Emulation.

Terminal emulation TE2000 AX via network

DEVICE = *ioadr*,*keyword*,"*socketnr[/t]*"[,*kennung*[,*timer*]]

This parameter defines the operation of a (new) TE2000 AX terminal emulation via network.

Terminal emulation TE-ALPHA via network

DEVICE = *ioadr*,*keyword*,"*socketnr[/t]*"[,IPKS[,*timer*]]

This parameter defines the operation of an (old) TE-ALPHA terminal emulation via network.

ioadr IO address of the device according to ORG generation (4-digit hexadecimal number)

keyword The following type specifications are permitted :
DS074NET / DS074FNET TE2000 AX/TE-Alpha via network
DS075NET / DS075FNET TE2000 AX via network

socketnr Socket number for the communication between the M2000 system and the terminal emulation. The socket number must be transmitted as start parameter when the remote terminal emulation is started. Socket port number according to administration.
Recommended range: >10000.
The established connection between M2000 and the terminal emulation will be time-monitored. If no data are exchanged within **t** seconds (default value: 180s), M2000 terminates the connection. The monitoring period ***t*** can be parameterized with the socket number (see examples).

kennung Enables the terminal emulation in the dongle.
IPKS identification (kennung): Enables the functions of the (old) TE-ALPHA/TE-DISIT terminal emulation in the central M2000 dongle.
TE2000 / TE2000FG identification (kennung): Enables the functions of the (new) TE2000-AX/-FG terminal emulation in the central M2000 dongle.

timer Time in ms the inputs/outputs are delayed to accommodate slow visual display units (e.g., ES05) and user programs.

Examples

```
DEVICE=8031,DS074FNET,"20011";
DEVICE=A301:3,DS075NET,"20011";
DEVICE=A308:3,DS074NET,"20011"      ;default time monitoring
DEVICE=A308:3,DS074NET,"20011/0"    ;no time monitoring
DEVICE=A308:3,DS074NET,"20011/3600" ;time monitoring 1h
```

Table 1 Display unit types and keywords for *mpar.sys* parameter definition

Display unit	**Keyword**		
Type	Serial connection	Local terminal	Network terminal
3974R	DS075	—	—
3974MT	3974MTSER	—	—
DS074/75	DS075	—	—
DS078	DU78	—	DU78
DS075_DISIT	DS075_DISIT	DS075_DISITLOK	DS075_DISITNET
TE2000 FG	DS075	DS075LOK	DS075NET
TE2000 AX	DS075	DS074/DS075LOK	DS074NET/DS075NET
TE2000 AX	DS075F	DS074F/DS075FLOK	DS074FNET/DS075FNET
TE2000 DX	3974MTSER	3974MT	3974MTNET
TE2000 DX	3974MTFSER	3974MTLOK	3974MTNETF
TE-KONS	—	DS074	—

Note If two or more network cards are installed in the computer, the IP address (or surrogate name) of the corresponding card can be assigned to a specific visual display unit. The surrogate name (alias) must be entered in the *hosts* file (**systemdir***system32\drivers\etc\...*). The corresponding parameter line is structured as follows:

DEVICE=*ioadr,keyword, ipadr*:*port[/t],kennung*
DEVICE=*ioadr,keyword,aliasname:port[/t],kennung*

Example

```
DEVICE=8030,3974MTNETF,192.168.2.10:20010/0,IPKS;
```

```
DEVICE=8030,3974MTNETF,NetCard-LAN1:20010,IPKS;
```

The connection is established via network card 1.

```
DEVICE=8038,DS075FNET,192.168.2.13:20013/0,TE2000;
```

```
DEVICE=8038,DS074NET,192.168.2.13:20013/0,IPKS;
```

```
DEVICE=8038,DS075NET,NetCard-LAN2,20013,TE2000;
```

The connection is established via network card 2.

Note A detailed description is given in the User Manual:

TE2000 Terminal Emulation.

4.2.2 Storage Media

Disk drive occupation mode

`mode = share`

MODE specifies the occupation mode (exclusive/shared) of the disk drives. If the mode parameter is not set, `exclusive` mode will be active (default setting). The use of the MODE parameter allows you to access the disks from both system environments. `Shared` mode must be used if the disks of an emulated SICOMP system are accessed via the Windows program *sicview.exe* (see section Auxiliary Programs). This mode can also be used to save disks of an emulated SICOMP system under Windows without having to terminate the emulator. In this case make sure that all DVS files (for example) are closed, or that the user system is not started.

Disk drive **PS048, PS049**

`DEVICE = ioadr[-devnum],keyword,"pathname"[,flags]`

This parameter is used to define the simulation of a PS048/49 (disk drive) sub-device by means of a Windows file.

`ioadr` IO address of the device according to ORG generation (4-digit hexadecimal number)

`-devnum` Device number of the sub-device

`keyword` PS048 designates a device with 64640 sectors
PS049 designates a device with 25856 sectors

`pathname` Location of the corresponding Windows file (the complete path must be entered). If the file does not exist, it is created by the emulator in the defined length. The file can then be formatted and set up with INITM, or physically described with MCSAVE.

`flags` Definable parameter (hexa)
= 1 WRITE THROUGH
This parameter ensures that changed data is written to the disk and doesn't remain in the CACHE. This is a useful option for DVS data carriers.

Important! The parameter entries of the specified disks must be entered in the *mpar.sys* file in such a way that *ioadr-devnum* is sorted in ascending order.

Example

```
DEVICE=8200-0,PS048,"c:\sicomp\disks\SYSPS0.sic";
```

Disk drive **FP0xx**

DEVICE = *ioadr*[*-devnum*],*keyword*,*anzsek*,"*pathname*"[,*flags*]

This parameter is used to define the simulation of an FP023, FP024, ... disk drive partition by means of a Windows file.

ioadr IO address of the device according to ORG generation
(4-digit hexadecimal number)

-devnum Device number of the device

keyword Device designation. Permitted are:
FP023, FP024, FP025, FP026, FP030, FP0XX, FP0XXAE

anzsek Number of sectors (65535 max.)

pathname Location of the corresponding Windows file (the complete path must be entered). If the file does not exist, it is created by the emulator in the defined length. The file can then be formatted and set up with INITM, or physically described with MCSAVE.

flags Parameter definition (hexa)
= 1 WRITE THROUGH
(see above)

Important! The parameter entries of the specified disks must be entered in the *mpar.sys* file in such a way that *ioadr-devnum* is sorted in ascending order.

Example

```
DEVICE=8200-0,FP023,63492,"c:\sicomp\disks\DVSPS0.sic",1;
```

Conversion from
ASCII to EBCDIC
(and vise versa)

If a disk is defined as FP0XXAE, the system checks prior to writing whether bit 14 (SIBI) in PHYSPB word 3 (ADDIN1) is set. If this is the case, the data to be written are converted from ASCII to EBCDIC. Similarly, the system checks whether bit 14 (SIBI) in PHYSPB word 3 (ADDIN1) is set when data are read from the disk. If this is the case, the read data are converted from EBCDIC to ASCII.

Example

```
DEVICE=8100-0,FP0XXAE,64640,"c:\sicomp\disks\sysPS0.sic";
```

Floppy disk drive

DEVICE = *ioadr,*FLOP,"*pathname*"
DEVICE = *ioadr,*FLOP18,"*pathname*"

This parameter is used to define a floppy disk drive.

ioadr IO address of the device according to ORG generation
(4-digit hexadecimal number).

FLOP Operation of a floppy disk drive (3.5" 1.44MB or 5.25" 1.2MB) without compatibility to SICOMP M. On a computer that does not support FM modulation for track 0/page 0, this floppy disk drive can be used for data backup and the data exchange between several emulations. In the emulated system, the FLOP disk drive can be used like an FD044 drive (also in connection with INITM). However, a diskette written with FLOP (15 sectors format) cannot be read on a SICOMP M computer.
FLOP18 Similar as above but written in 18 sectors format.

pathname The pathname is :\\.\A:
\\.\B:

Example

```
DEVICE=8104,FLOP,\\.\A:;
```

Magnetic tape cassette - drive **MK82**

`DEVICE` `=` *`ioadr`*`,MK082,"`*`pathname`*`"`

This parameter allows you to operate an MK82 compatible magnetic cassette drive of the type Tandberg Data Panther TDC 3660/3820 for writing and reading 150 MB cassettes.
Magnetic tape cassettes of an MK80/81 device with 60MB can be read only.

`ioadr` IO address of the device according to ORG generation
(4-digit hexadecimal number)

`path name` Location of the corresponding Windows device

Important! If the MK82 is used with GBS, the **IO address 8106** must be specified.
In this case it may be necessary to create different configuration files (*mpar.sys*) for the user system and for GBS operation.

A magnetic tape cassette drive generated in ORG-M must be defined in the MPAR.SYS file even if no magnetic tape cassette drive is connected to the computer. Only then it is ensured that read and write requests to device MKSK*m.n* will be rejected with an error message. Otherwise, the emulator would interpret the transmission attempt as a request to a disk, which could lead to errors such as the abortion of the disk process.
A software solution for this problem is not available.

Example

```
DEVICE=8106,MK082,\\.\TAPE0;
```

Optical disk **OP31**

`DEVICE` `=` *`ioadr`*`,OP31,"`*`pathname`*`"`

This parameter defines the emulation of a rewriteable OP31 optical disk via the SONY RMO-S580 drive.

`ioadr` IO address according to ORG generation
(4-digit hexadecimal number)

`pathname` The pathname is "\\.\PhysicalDrive%".
% is the disk number in Windows, starting with 0 (see Windows hard disk manager)

Example

```
DEVICE=8202,OP31,\\.\PhysicalDrive1;
```

Note Only one DEVICE ... line is entered in the parameter file for every optical disk drive connected to the computer (sub-devices are not used).
see also OP31 Emulation

4.2.3 Printers

Printer operation, **parallel port**

`DEVICE` `=` *`ioadr`*`,DRUA,"`*`pathname`*`"[,`*`flags`*`[,`*`timer`*`]]`

This parameter diverts the *DRUA* device to the printer connected to the parallel interface.

ioadr IO address according to ORG generation
(4-digit hexadecimal number)

pathname Pathname of the corresponding Windows device, usually LPT1

flags Definable parameter (hexa)
= 0

timer WRITE monitoring period in seconds (default value: 20 s)

Example

```
DEVICE=8020,DRUA,"LPT1";
DEVICE=A300:3,DRUA,"LPT1";
```

Important! No printer driver should be installed or activated for the parallel interface in Windows.

Printer output via **Windows printer driver**

`DEVICE` `=` *`ioadr`*`,DRSPOOL,"`*`pathname`*`"[,timer]`
`DEVICE` `=` *`ioadr`*`,DRSPOOLF,"`*`pathname`*`"[,timer]`

This parameter is used to pass on the DRUA device to a Windows printer driver. The first output opens a document, which is closed after a default period of 5 seconds if no subsequent outputs follow. A new document is opened after the default period of 5 seconds has elapsed.
DRSPOOLF furnishes the outputs with the trailing control character FF (FormFeed) required by some laser printers.

ioadr IO address according to ORG generation
(4-digit hexadecimal number).

pathname Pathname of the corresponding Windows device
(local or network)

timer Period in milliseconds after which a document is closed again. If the parameter is not defined, the default value of 5000 ms is active.

Examples

```
DEVICE=8020,DRSPOOL,"LPT1";
DEVICE=8020:1,DRSPOOL,"LPT1",1000;
DEVICE=8020,DRSPOOL,"\\DRSERVER\DRUA00",7000;
DEVICE=8020,DRSPOOLF,\\192.168.1.7\DRUA10;
```

Note The following applies to network printer outputs:

`DRSERVER`	= Name of the printer server
`192.168.1.7`	= IP address of the printer server
`DRUA00`	= Local enable name of the network printer
`DRUA10`	= Local enable name of the network printer

Printer output to
Windows file

`DEVICE` = *`ioadr`*`,DRUA,"`*`pathname`*`"`

This parameter is used to transfer the printer output to a Windows file. The output starts at the beginning of the file. An existing file will be deleted.

`ioadr` IO address according to ORG generation (4-digit hexadecimal number).

`pathname` Complete path of the windows file to which the output is transferred. The file is set up implicitly.

Examples

```
DEVICE=8060,DRUA,"..\Printer\drua0.txt";
DEVICE=8060:3,DRUA,"..\Printer\drua0.txt";
```

Printer output to
Windows file

`DEVICE` = *`ioadr`*`,DRSPOT,"`*`pathname`*`",`*`timer`*
`DEVICE` = *`ioadr`*`,DRSPOTS,"`*`pathname`*`",`*`timer`*

Note

DRSPOT creates DOS compatible file names
(names > 8 characters, suffix > 3 characters, e.g., 19980610123007.8028)
(suffix .8028 = IO addr (pseudo) printer in *mpar.sys*).
DRSPOTS creates DOS compatible file names
(names <= 8 charcaters, suffix <= 3 characters, e.g., 10123007.13)
(suffix .13 = device index = line no. of the device entry in *mpar.sys*).

This parameter transfers the printer output to a Windows file. A file with the name TTSSMMss.iii (*day, hour*, *minute, second.device index,* in *mpar.sys* /DRSPOTS/) or JJJJMMTTSSMMss.ioa (*year, month, day, hour, minute, second.io address /DRSPOT/*) is created. A timer is started after every printer output to the file. After the timer period has elapsed, the file is closed. A subsequent printer output is transferred to a new file.

`ioadr` IO address according to ORG generation (4-digit hexadecimal number).

`pathname` The complete path. The file is set up implicitly.

`timer` Period in seconds after which the file is closed. Subsequent printer outputs are transferred to a new file.

Example

```
DEVICE=A358:3,DRSPOTS,"I:\Printer\Nsu",30;
```

Printer output to **Windows file** (parameter-controlled)

`DEVICE` = *`ioadr`*,`DRSPOTX`,"*`file`*"

DRSPOTX defines the parameter-controlled printer output to a Windows file.

`ioadr` IO address according to ORG generation (4-digit hexadecimal number).
`file` (Path)name of the parameter file.

Note The following rules must be observed when a parameter file is created:
Comments start with a semicolon.
The parameter names SUPPRESS, CHANGE, START, STOP may be used up to 255 times, a string can have a maximum length of 64 charcaters, all characters can be used.
All other parameter names may only be used once (if a name is used more than once, always the last definition applies).
In character strings, special characters can be represented as hexa patterns (see below):
The two characters following a backslash (\) are interpreted as hexa pattern. The \ character itself can only be represented as \5d.
The **"** character can only be represented as \22.
Only <u>one</u> START or STOP string will be interpreted in every output record.
If the parameter file is changed during running operation, the changes will become effective after the emulator has been restarted, or after the "Reset device" function has been activated by the TW.EXE application (open files will be closed before the reset function is executed).
Time-controlled one-time or reoccurring processes must be implemented with the Windows command **at** *time command.* This requires that Windows SCHEDULE+ is started (**net start** schedule).

Example **at** 23:00 netcopy.bat

Dynamic functions are processed in the following sequence:
1. SUPPRESS: all characters to be deleted are removed
2. CHANGE: the necessary replacements after deletion are carried out.
3. START: checks whether the data set contains a 'start character sequence' <u>after</u> the deletion and replacement.
4. STOP: checks whether the data set contains a 'stop character sequence' <u>after</u> the deletion and replacement.

Parameters:

PATH [*drive*:[*path*]] ; Location where the files to be created are stored (default: M2000 start directory)
SPOTTIME *sec* ; Definition of the monitoring time (default: 0, no monitoring time specified).
DOSNAMES ; Use of DOS compatible file names,
NTNAMES ; Use.of file names not compatible with DOS (default).
LFDNAMES ; File names are created in the form of 'consecutive numbers'
STARTNAMES *n* ; *n* characters following START "*string*" form the beginning of the file name (z...z_JJJJMMTThhmmss.ioadr)
EXTENSION x....x ; Freely definable (file) extension
SHARE ; The exclusive allocation status of the output files is canceled.
SUPPRESS "*string*" ; Characters (sequences) are removed prior to output
CHANGE "*string1*" "*string2*" ; Characters (sequences) are replaced prior to output,
*string*1/*string2* can have different lengths.
START "*string*" ; Definition of a 'start character sequence'
STOP "*string*" ; Definition of a 'stop character sequence'
WINDOW "*name*" ; outputs also shown in window name

```
WINDOWT "name"                ; as above, but with preceding time stamp
COMMAND "[drive:[path]] file" ; Definition of a command;
```

The command is transmitted after the target file is closed. 5 parameters can be specified:

The 1st parameter designates the currently closed file,
the 2nd parameter contains "file closing event" identifications such as:

"to"	*Period*
"aa"	*Start character sequence*
"zz"	*Stop character sequence*
"bo"	*Boot ORG* or reset printer
"nt"	after termination of the emulation

The 3rd parameter specifies the directory where the file is stored,
the 4th parameter specifies the name of the currently closed file,
the 5th parameter specifies the extension of the currently closed file.

Example

```
DEVICE=A358:3,DRSPOTX,"\sicomp\spotx.par";
```

The parameter file *spotx.par* could look as follows:

```
PATH e:\temp                                                          ; ( 1 )
DOSNAMES                                                              ; ( 2 )
EXTENSION DRUA1                                                       ; ( 3 )
SPOTTIME 20                                                           ; ( 4 )
SUPPRESS "\0D\0A\03"                                                  ; ( 5 )
CHANGE "\0D\0A" "\0A"                                                 ; ( 6 )
START "//JOB:"                                                        ; ( 7 )
STOP   "//JEND"                                                       ; ( 8 )
STARTNAMES 7                                                          ; ( 9 )
SHARE                                                                 ; ( 10 )
COMMAND "e:\temp\drspotx.bat" "drfile1" "to" "\temp" "drfile1" "txt"  ; ( 11 )
```

(1) The print files are stored in the directory *e:\temp*.

(2) DOS compatible file names are used.

(3) The freely definable (file) extension is .DRUA1 instead of *.ioadr* .

(4) The print files are closed if no output is initiated within the period of 20 seconds.

(5) The character sequence CR-LF-ETX is suppressed.

(6) The character sequence CR-LF is replaced by LF.

(7) The current print file is closed as soon as the character sequence "//JOB:" is detected; a new file is created and "//JOB:" is transferred to the new file.

(8) The current print file is closed as soon as the character sequence "//JEND" is detected and the string "//JEND" is written to the end of the file that is closed. Then a new print file is opened to which data are written until the output ends (as defined).

(9) The first 7 characters following "//JOB:" (e.g., JOBCOPY), form the beginning of the new file name, e.g., *JOBCOPY_20020812135801.0202*.

(10) The (file) status *shared allocation* is activated, the file can be viewed during output.

(11) The print file *drfile1.txt* is closed after the period (SPOTTIME) has elapsed (identification *to*). After the file is closed, *drspotx.bat* is activated.

Note If SPOTTIME is set to 0 (default value), the file created last remains open and "receptive" until another event causes the file to close (START ..., STOP ..., etc.).

Printer operation (serial) with **DR202**

`DEVICE` `=` *`ioadr`*`,DR202,"`*`pathname`*`"[,`*`flags`*`[,`*`timer`*`[,`*`verz1`*`]]]`

This parameter allows you to operate a DR202 printer connected to the serial interface.

`ioadr` IO address according to ORG generation (4-digit hexadecimal number).

`pathname` Designates the corresponding serial interface in Windows (COMx or \\.\COMxx).

For DR202, the serial interface can also be on a remote computer. The *pathname* is specified in the form “REM:x“. Parameter x denotes the port number of the first of two TCP/IP sockets created by the device process on the M2000 computer.

(see section Interface Concentrator)

`flags` Defined parameters (hexa)
= 1 Device inoperable detection for TTY interface
(jumper between RxD and CTS required)
= 2 DSR scan prior to output to V.24 interface
= 4 ready/busy

`timer` WRITE monitoring period in seconds (default value = 20 s)

`verz1` Time delay in ms before data transmission is started (if the connected device is too slow).

Note During parameter definition via ORG (/PARAM ..), the M2000 driver process evaluates bytes 0,1,2,3,4.

Examples Printer connection to the local COM port:

```
DEVICE=8060,DR202,"COM7";
DEVICE=8060:1,DR202,"COM7";
```

Printer connection to the COM port of a remote Windows system:

```
DEVICE=8060,DR202,"REM:23000";
```

Printer operation (serial), **all printers**

DEVICE = *ioadr*,DRCOM,"*pathname*"[,*typflags*[,*time1*[,*time2*]]]

This parameter allows you to operate all printers connected to the serial interface.

ioadr IO address according to ORG generation
(4-digit hexadecimal number).

pathname Designates the corresponding serial interface in Windows (COMx or "\\.\COMxx).

For DRCOM, the serial interface can also be on a remote computer. The *pathname* is specified in the form “REM:x“. Parameter x denotes the port number of the first of two TCP/IP sockets created by the device process on the M2000 computer.

(see section Interface Concentrator)

typflags Parameters (hexa):
printer type ‘’typ’’,
string ‘’CRLF’’,
or flags:
= 1 Break detection for TTY
= 2 Break detection for V.24
= 4 Ready/busy
= 8 Do not define serial port parameters - the MODE command applies
= 10 Similar to parameter(string) CRLF
= 20 Ignore SICOMP parameters (/PARAM)

time1 WRITE monitoring period in seconds (default value = 20 s)

time2 Time delay in ms before data transmission is started (if the connected device is too slow).

Note During parameter definition via ORG (/PARAM ..), the M2000 driver process evaluates bytes 0,1,2,3,4.

Example Printer connection to the local COM port:

```
DEVICE=8060,DRCOM,"\\.\COM7",8,20,100;
DEVICE=8060:1,DRCOM,"COM7";
```

Printer connection to the COM port of a remote Windows system:

```
DEVICE=8060,DRCOM,"REM:23000";
```

Printer output to **Windows** window

DEVICE = *ioadr*,DRWIN,"*window*"[,"*file1*","*file2*","size"]

The printer output is transmitted to a Windows window and can also be stored in files.

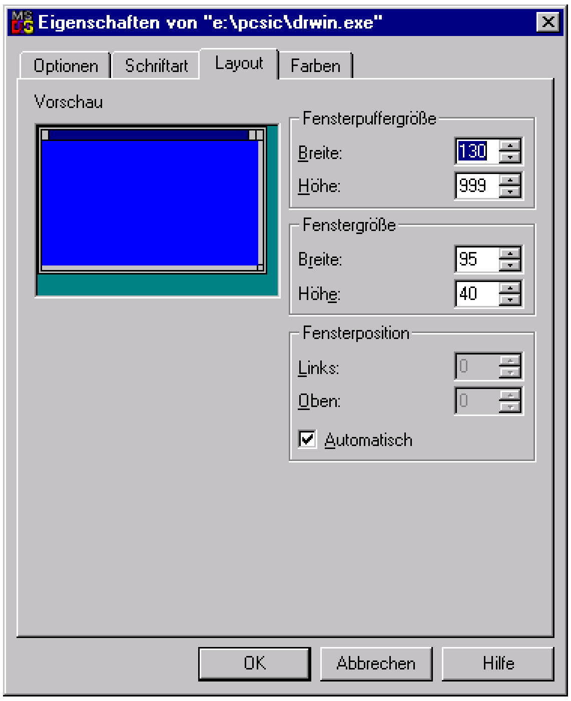
ioadr	IO address according to ORG generation (4-digit hexadecimal number)
window	Window name
file1	1st log file
file2	2nd log file
size	Max. file size in kilobytes

Examples

```
DEVICE=8060,DRWIN,"DRUA0","outfile1","outfile2","10";
DEVICE=A360:1,DRWIN,"DRUA0";
```

The DRWIN parameter diverts the printer outputs to a Windows window. After the start of the emulation, the screen displays an additional window named DRUA0. This window can be configured. Right-click the header of the DRUA0 window and select *Properties.* Select *Layout* ⇒ *Window buffer size* - Window size and specify the number of printer lines the window shall be capable to buffer (e.g., 999). The window now works like a cyclic buffer with a size of 999 lines.

Now, to transmit parts of the printer outputs (for example, error messages) to the computer's printer, proceed as follows:
Right-click the header and select *Edit* ⇒ *Select*. Select the lines you want to print. Copy the selected area to the clipboard (CTRL+C). Open a text editor, such as WRITE, and paste the selected area into a document (CTRL+V). Then print out the document.



Fensterpuffergröße
Breite: 130
Höhe: 999

Fenstergröße
Breite: 95
Höhe: 40

Fensterposition
Links: 0
Oben: 0
☑ Automatisch

(Additional) output to log files

Additional output to log files can be implemented with the parameters *file1*, *file2* and *size* (see example 1).

The printer output is transferred to *file1* first; if *file1* has reached maximum size (*size**1024), the printer output is transferred to *file2*; if *file2* has reached maximum size, the printer output is stored in *file1* after the old contents has been deleted.

Always a complete printer line is written to the file before the file is scanned for maximum size.

The current settings are stored in the *window.par* (e.g., *DRUA0.par*) when the emulation is terminated.

If the emulation is restarted and the parameters haven't been changed, the printer outputs are added to the log file written last followed by the same procedure as described above.

If the parameters have been changed or if problems are encountered opening the file written last, the printer outputs are transferred to a new *file1*.

A message line (--End DRWIN--) is added at the end of the current log file when the procedure is terminated.

Printer output with date/time to a **Windows window**

```
DEVICE = ioadr,DRWINT,"window"
```

The DRWINT parameter is similar to the DRWIN parameter; date and time are added at the beginning of each line of the printer output in the following format: `YYYY-MM-DD hh:mm:ss.ms`. This format is ideal for message printers.

ioadr IO address according to ORG generation (4-digit hexadecimal number)

window Window name

Example

```
DEVICE=8060,DRWINT,"DRUA";
```

DRWINT *example*

```
2002-08-1311:58:02.701 CDLI TO DRUA
2002-08-13 11:58:02.701 TESTCD 28 CA CBUET 65271 42
```

Note Entering `DRWINMIN/DRWINTMIN` instead of `DRWIN/DRWINT` minimizes the printer windows.

4.2.4 Interface Connections

DU02 computer interfacing Gateway function

`DEVICE`=*ioadr*,`DU02`,`"SOCKET"`,`"`*host*`"`,*port, type,*0,*size,*[*t1*[,*t2*[,*t3*]]]

This parameter activates the emulation of a DU02 computer interface connection (fiber optics interface connection).
It is thus possible to connect two emulator systems.

`ioadr` IO address according to ORG generation (4-digit hexadecimal number).

SOCKET For TCP/IP transmission via Win socket

`host` Host name or empty string (see 6.2.2 DU05 communication to TCP/IP)

`port` PortNumber (10000-plus) (see 6.2.2 DU05 communication to TCP/IP)

`type`

B:	balanced
UP:	unbalanced primary
US:	unbalanced secondary

`size` Maximum size of the replaced user data sets.

`t1` Waiting time in milliseconds after connection has been established (default value: 10000 ms)

`t2` Max. waiting time for ORG input command (default value: 5000 ms)

`t3` Max. waiting time for acknowledgement from peer system (default value: 10000 ms)

Example

```
DEVICE=8300,DU02,"SOCKET","WINNT1",22220,US,0,1024,30,5,0;
```

active side at connection buildup; enter host name

```
DEVICE=8300,DU02,"SOCKET","",22221,US,0,1024,30,5,0;
```

passive side at connection buildup; empty string

DU03 computer interfacing

The generation of a DU03 module required four device names (sub-devices) in ORG-M (see below). The individual devices operate different channels with different functionalities (e.g., *ISO line, SNVS line*).

Example: Generation of a single DU03 in ORG-M:

```
/G:9400,KHDK0011=DU003,ST;
/G:9402,KHDK0012=DU003,ST;
/G:9404,KHDK0013=DU003,ST;
/G:9406,KHDK0014=DU003,ST;
```

The details are not important for the user of the emulation as always all generated sub-devices are emulated.

```
DEVICE = ioadr:255,DU03,"NDIS",0;       (WinNT)
DEVICE = ioadr:255,DU03,"NDIS2000",0;   (Win2000)
```

`ioadr` IO address according to ORG generation(4-digit hexadecimal number)
`:255` Fixed value for masking the sub-device address

`NDIS` The DU03 emulation runs under WinNT

`NDIS2000` The DU03 emulation runs under Win2000

The DU03 emulation completely maps the original -Data transfer- functionality. Test and maintenance jobs are not mapped but will be processed and completed without indication for compatibility reasons.
The operation of an DU03 emulation in a highly burdened network may result in loss of performance of the overall emulation. In this case, the use of switches, bridges, etc., even of a multiprocessor computer, can help improve network performance.
The DU03 computer interfacing emulation requires no special hardware - all that is required is a common network card (an emulated DU03 can thus be operated with 100 Mbit/s). On the emulation computer, a packet or NDIS driver must be installed that IPKS can presently supply for WindowsNT and Windows2000. The driver enables the DU03 emulation to read and check the relevance of the received data packets via the corresponding network card, and to transmit data packets.

The following is required for the operation in WinNT/Win2000:

- Packet driver software for the old NDIS driver (WinNT) or
- software for the new Windows2000 packet driver
- Installation according to the information in the supplied *packet.htm*
- The parameter file *mpar.sys* distinguishes between NDIS(old) and NDIS(new)

Examples

```
device = 9400:255,DU03,"NDIS",0;
device = 9400:255,DU03,"NDIS2000",0;
```

The IO address *(ioadr)* must always be the address ending with '00' (device=9400:255).

DU04 computer interfacing via DF32/42

DEVICE = *ioadr*,DU04,"*pathname*",*dfadr*[,*buflen*]
ioadr,DU04,"*pathname*",*dfadr*[,*buflen*]

This parameter defines the emulation of a DU04 computer interface connection via DF32/DF42 module interface. The protocol driver COMDRV (3964R) must be installed for the DF32/DF42 module.
In this case the DF32/42 processes the 3964R protocol so that the description and parameter definition of this module applies.

ioadr IO address according to ORG generation
(4-digit hexadecimal number).

pathname Pathname of the corresponding Windows device.
Operation only via DF32/42. The interface must be loaded and defined with the corresponding board software (COMDRV driver) using Comsoft software aids (see DFxx Manual). The interface settings are not changed by the emulation.

dfadr The board interface is addressed in the form:

bs

b	board number 0..4
s	interface number in board 0..7

buflen Receive buffer length in bytes for the DF32/42 driver. This length must coincide with the value specified for the DF32/42 protocol driver installation.
Default value: 1024.

Examples

```
DEVICE=8060,DU04,"\\.DFXX0",13;
```

DU04 computer interfacing via any serial interface

DEVICE = *ioadr*,DU04,"*pathname*",*par*
DEVICE = *ioadr*,DU04S,"*pathname*",*par*
This parameter defines the emulation of a DU04 computer interface connection via any serial interface. The serial interface must be specified as COMxx under Windows. Emulator version V3.06 K025 and higher offers an additional device (**DU04S)** that solves the problems resulting from too short waiting times. The letter 'S' stands for 'slow', i.e., the emulator increases the internal acknowledgement times.

ioadr IO address according to ORG generation
(4-digit hexadecimal number).
pathname Pathname of the corresponding Windows device interface, i.e. COMx or \\.\COMxx. The interface must be predefined with the Windows command *mode* (default setting). Overparameterization through ORG-M is implemented to a large extent.
par Protocol parameters ***wxpb***:

w	0 = 4 kB internal buffer (default value) 1 = 64 kB internal buffer
x	0 = no BREAK detection 1 = TTY BREAK detection
p	0 = low priority 1 = high priority
b	0 = without BCC 1 = with BCC

Example

```
DEVICE=8060,DU04,"COM5",1011;
DEVICE=8060,DU04S,"\\.\COM15",11;
```

DU04 computer interfacing via TCP/IP

```
DEVICE = ioadr,DU04,"SOCKET",port-own,
         host-remote:port-remote;
```

This parameter defines the emulation of a DU04 computer interface connection via TCP/IP and LAN or RAS. A LAN / RAS network adapter and TCP/IP must be installed in Windows (see also DU05 Computer Interfacing).

ioadr	IO address according to ORG generation (4-digit hexadecimal number).
port-own	Port number/socket number in the local computer (e.g.: 22223).
host-remote	Host name or address in the remote computer (e.g.: winnt1 or 220.250.1.122).
port-remote	Port number/socket number in the remote computer (e.g.: 22225).

Examples

```
DEVICE=8060,DU04,"SOCKET",22223,winnt7:22225;
DEVICE=8060,DU04,"SOCKET",22223,220.250.1.155:22225;
```

DU05 computer interfacing via DF42 Parameter definition via emulator

```
DEVICE = ioadr,DU05, "\\.\DFXX0",frame,parfile
```

This parameter defines the emulation of a DU05 computer interface connection via DF42 module interface.
The protocol driver MSV2 must be installed for the DF42 module. In this case the DF42 processes the MSV2 protocol so that the description and parameter definition of this module applies.
Alternatively, the DU05 interfacing can be diverted to a TCP/IP interface connection via the Win socket interface (see example on the following page).
In addition to this device entry in the *mpar.sys* configuration file, the DU05 emulation requires a parameter file (*parfile*) that allows you to define additional operating modes. Please refer to the description in section 6.2.

ioadr	IO address according to ORG generation (4-digit hexadecimal number).
frame	Maximum FRAME length
parfile	Path and name of the parameter file acc. to 6.2 (for DFXX0)

Examples

```
DEVICE=8300,DU05,"\\.\DFXX0",512,c:\sicomp\du05.par;
```

DU05 computer interfacing via DF42 Parameter definition via DFTEST

```
DEVICE = ioadr,DU05, "\\.\DFXX0",board,unit,frame,options
```

This parameter defines the emulation of a DU05 interface connection via DF42 interface. The protocol driver MSV2 must be installed for the DF42 module. In this case the DF42 processes the MSV2 protocol so that the description and parameter definition of this module applies.

ioadr	IO address according to ORG generation
board	Board number of the Comsoft module
unit	Interface number of the Comsoft module
frame	Maximum FRAME length
options	Presently always 0

Examples

```
DEVICE=8300,DU05,"\\.\DFXX0",0,0,1024,0;
```

DU05 computer interfacing Gateway function

```
DEVICE=ioadr,DU05, "SOCKET","host",port[/t0],size,
       [t1[,t2[,t3]]]
```

This parameter diverts the emulation of a DU05 computer interface connection to a TCP/IP connection via the Win socket interface. A standard LAN connection is used.

ioadr IO address according to ORG generation
(4-digit hexadecimal number).

"SOCKET" for TCP/IP transmission via Win socket

host Host name or empty string (see 6.2.2 DU05 Communication to TCP/IP)

port Port no. (10 000... plus) (see 6.2.2 DU05 Communication to TCP/IP)

t0 Connection monitoring time in seconds (default setting: no monitoring)

size Max. size of the replaced user data sets

t1 Waiting time after connection buildup in milliseconds
(default value: 10000 ms)

t2 Max. waiting time for ORG input command (default value: 5000 ms)

t3 Max. waiting time for acknowledgement from peer system
(default value: 10000 ms)

Example

```
DEVICE=8300,DU05,"SOCKET","HOST1",22220,512,20000,10000,30000;
```
active side at connection buildup, enter host name
```
DEVICE=8300,DU05,"SOCKET","",22221,512,20000,10000,30000;
```
passive side at connection buildup; empty string

Note

If two or more network cards are installed in the computer, you can use the IP address (or surrogate name) of the corresponding card to establish the connection. The surrogate name (alias) must be entered in the *hosts* file (**systemdir**\system32\drivers\etc\...). The corresponding parameter line is structured as follows:
DEVICE=*ioadr,*DU05, "SOCKET","[partner]-*ipadr*",*port,size*
DEVICE=*ioadr,*DU05, "SOCKET","[partner]-*aliasname*",*port,size*

Examples

```
DEVICE=B400,DU05,"SOCKET","peer-11.12.13.1",20000,2048;
```

```
DEVICE=B400,DU05,"SOCKET","peer-LAN1",20000,2048;
```
The connection is established via network card 1.

```
DEVICE=B400,DU05,"SOCKET","peer-11.12.13.2",20000,2048;
```

```
DEVICE=B400,DU05,"SOCKET","-11.12.13.2",20000,2048;
```

```
DEVICE=B400,DU05,"SOCKET","peer-LAN2",20000,2048;
```
The connection is established via network card 2.
.

DU06 computer interfacing via DF32/42

`DEVICE` = *ioadr*,`DU06`, "*driver*",*kenn1*,*kenn2*,*type*,0,*frame*,*t1*,*t2*,*t3*,*parfile*

This parameter defines the emulation of a DU06 computer interface connection via DF32/DF42 module interface. The protocol driver HDLC UNC-2 or HDLC BAC 2.8 must be installed for the DF32/DF42 module. In this case the DF32/42 processes the HDLC protocol so that the description and parameter definition of this module applies.
In addition to this device entry in the *mpar.sys* configuration file, the DU06 emulation requires a parameter file (*parfile*) that allows you to define additional operating modes. Please refer to the description in section:
6.3 DU06 Computer Interfacing.

`ioadr` IO address according to ORG generation
(4-digit hexadecimal number).

`driver` "\\.\DFXX0" for COMSOFT DFxx

`kenn1` DFxx module number

`kenn2` Interface number of the DFxx module

`type`
B: balanced
UP: unbalanced primary
US: unbalanced secondary

`frame` Max. I-FRAME length

`t1`
0: No monitoring
1: Input monitoring 1 second (message)
5: Input monitoring 5 seconds (message)
24: Input monitoring 24 seconds (message)
101: Input monitoring 1 second (message/terminate connection)
105: Input monitoring 5 seconds (message/terminate connection)
124: Input monitoring 24 seconds (message/terminate connection)

`t2`
0: Don't terminate dial connection during transmission pauses
1: Terminate dial connection during transmission pauses

`t3` Acknowledgement monitoring time in milliseconds

`parfile` Path and name of the parameter file acc. to 5.3

Note At present, 0 must be set for timers t1, t2 and t3!

Example DEVICE=8300,DU06,"\\.\DFXX0",0,0,US,0,0,0,0, g:\sicomp\du06us.par ;

DU06 computer interfacing

Gateway function

DEVICE=*ioadr*,DU06,"SOCKET","*host*",*port*,*type*,0,*size*,[*t1*[,*t2*[,*t3*]]]

This parameter diverts the emulation of a DU06 computer interfacing
to a TCP/IP connection via the Win socket interface.
A standard LAN connection is used.

ioadr IO address according to ORG generation
(4-digit hexadecimal number).

SOCKET Win socket for TCP/IP transmission

host Host name or empty string (see 6.4.4 DU06 Communication to TCP/IP)

port Port no. (10 000... plus) (see 6.4.4 DU06 Communication to TCP/IP)

type B: balanced
UP: unbalanced primary
US: unbalanced secondary

size Max. size of the replaced user data sets

t1 Waiting time in milliseconds after connection buildup error
(default value: 10000 ms)

t2 Max. waiting time for ORG input command (default value: 5000 ms)

t3 Max. waiting time for acknowledgement from peer system
(default value: 10000 ms)

Example

```
DEVICE=8300,DU06,"SOCKET","WINNT1",22220,B,0,1024,30,5,0;
```

active side at connection buildup, enter host name

```
DEVICE=8300,DU06,"SOCKET","",22221,B,1024,30,5,0;
```

passive side at connection buildup, empty string

Note

If two or more network cards are installed in the computer, you can use the IP address (or surrogate name) of the corresponding card to establish the connection. The surrogate name (alias) must be entered in the *hosts* file (**systemdir***system32\drivers\etc\...*).
The corresponding parameter line is structured as follows:
DEVICE=*ioadr,*DU06, "SOCKET","[partner]-*ipadr*",*port,type,0,size*
DEVICE=*ioadr,*DU06, "SOCKET","[partner]-*aliasname*",*port,type,0,size*

Example

```
DEVICE=B400,DU06,"SOCKET","peer-11.12.13.1",20000,B,0,1024;
```

```
DEVICE=B400,DU06,"SOCKET","peer-LAN1",20001,B,0,1024;
```

The connection is established via network card 1.

```
DEVICE=B400,DU06,"SOCKET","peer-11.12.13.2",20000,B,0,1024;
```

```
DEVICE=B400,DU06,"SOCKET","-11.12.13.2",20000,B,0,1024;
```

```
DEVICE=B400,DU06,"SOCKET","peer-LAN2",20001,B,0,1024;
```

The connection is established via network card 2.

KS100 communications-control

DEVICE = *ioadr*,KS100,*index*,*ptime*,*pmax*,*0*

This parameter defines the emulation of a KS100 communication via SINEC H1. A CP1413/CP1613 module must be installed and set up. KS100 runs under SIMATIC NET.

ioadr IO address according to ORG generation (4-digit hexadecimal number).

index Consecutive number (1 or 2) of the CP1413/CP1613 acc. to Windows parameter definition

ptime Waiting time in milliseconds for awaiting the results of synchronously executed CALL_INA calls
default value: 50 ms (for ptime = 0)

pmax Max. cycles for awaiting the results of synchronously executed CALL_INA calls
default value: 100 attempts (for pmax = 0)

Example

```
DEVICE=8060,KS100,1,0,0,0;
```

CS275
communication
control

DEVICE = *ioadr*,CS275,*0*,*1*

This parameter defines the connection to the TELEPERM M bus system CS275. A NAT module must be installed and set up.

ioadr IO address according to ORG generation (4-digit hexadecimal number).

The other parameters are fixed parameters.

Example

```
DEVICE=8060,CS275,0,1;
```

CP1400
communication
control

DEVICE = *ioadr*,CP1400,*par1*,*par2*

This parameter defines the emulation of a (CP1400) SINEC H1 interface connection via ISO protocols 1-7. A CP1413/CP1613 module must be installed and set up (see also CP1400 Communication Control).

ioadr IO address according to ORG generation (4-digit hexadecimal number).

par1 CP1413/CP1613 module number (1 or 2)

par2 0

Example

```
DEVICE=8060,CP1400,1,0;
```

UCP-2
communication processor

DEVICE = *ioadr*,UCP-2,*prsize*,*blsize*,*opt*

This parameter defines the emulation of an UCP-2 communication processor (i.e., of a TCP/IP communication). The device emulation diverts the communication connections to the Windows socket interface.

ioadr IO address according to ORG generation
(4-digit hexadecimal number).

prsize Minimum size (in kilobytes) of the device process in Windows
(e.g.,: 2048). You can use this parameter to reserve buffer areas in the device process in order to improve performance.
0 = no minimum size specified

blsize Max. transmit buffer size in kilobytes for SENDIT call (total of all sub buffer lengths). Default value: 4 kB (for blsize = 0). The other calls use the lengths transmitted by the SICOMP system.

opt
0 No options

1 Set up sockets with SO_KEEPALIVE and SO_REUSEADDR

2 Start processing thread with the same priority as the main thread

4 For socket task RECEIVEIT, buffer size **32768** is used at the interface to Windows instead of the buffer length specified by the user program.
The data to the user program are transmitted in portions in line with the buffer length specified by the user program.
This increases performance of specific applications, such as the FTP transfer of very large files.

Notes

- Values of parameter *opt* are interpreted in hexadecimal form, the bit patterns can be linked by an OR operation.
- Firmware load calls can be ignored in the emulation (otherwise the firmware data received by the emulation is ignored).
- If FTP is used (SICOMP programs UXFTP and UXFTPD), the FTP protocol in the Windows network must not be activated under the standard port numbers.
- If standard services are used, these must be commented out in the file *%systemroot%\drivers\etc\services*, or assigned with a number other than the standard port number to make sure that the service is not activated in Windows but in ORG only.

Example: If Remote Login (SICOMP program RDATR) is used, the line 'login 513/tcp' in file *%systemroot%\drivers\etc\services* must be commented out.

Examples

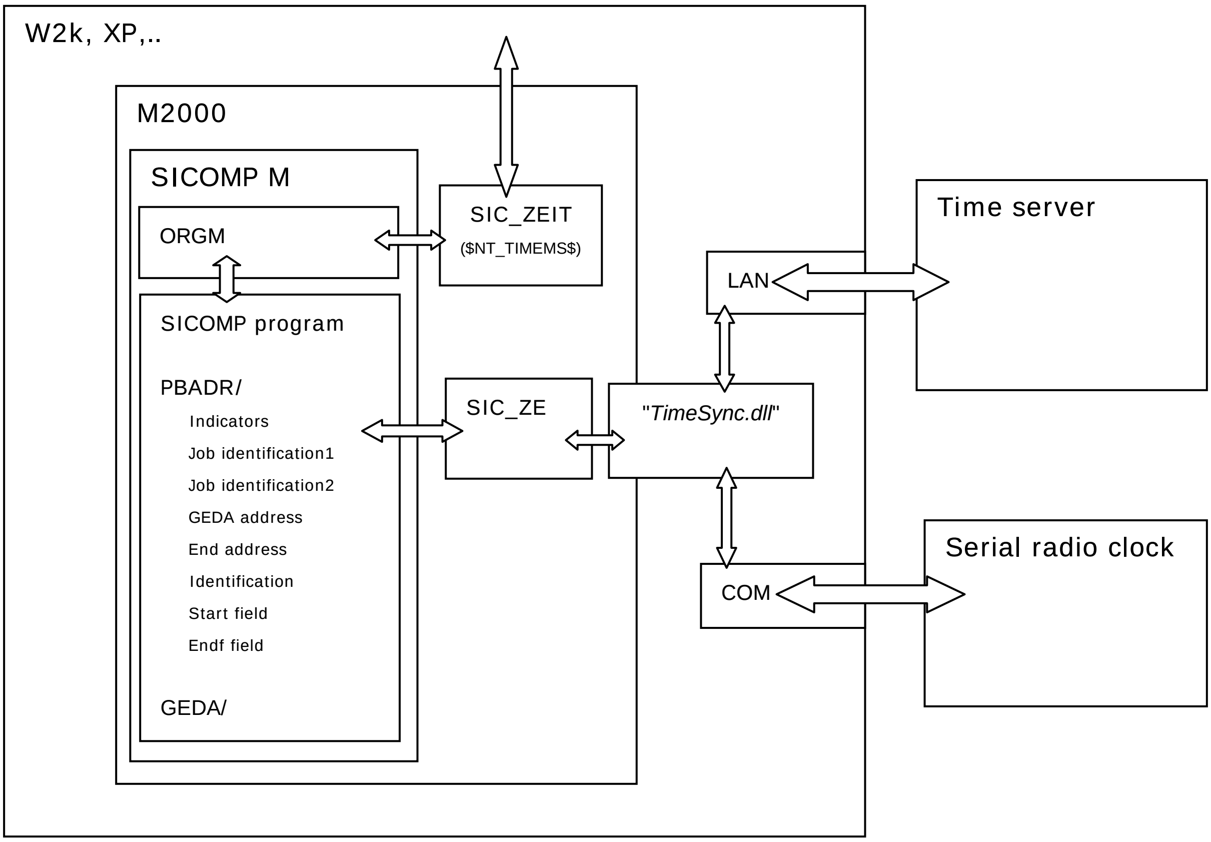
```
DEVICE=8070,UCP-2,0,0,0;
DEVICE=8070,UCP-2,2048,0,0;
```

4.2.5 Additional Devices

4.2.5.1 Time Synchronization

Time synchronization

The TIMESYNC parameter allows you to synchronize the emulated system with third-party systems (in our case with the Windows system) by means of a radio clock or a time server. An independent emulator process determines the current time via a DLL (*TimeSync.dll*) on the Windows level and makes this time available to a SICOMP program. Using the ORG call '$DATE', the SICOMP time and the summer/winter time identification are set in ORG. The emulator then updates the Windows time with the ORG time.
The time synchronization feature is described in detail in a separate document (TimeSync.pdf in **CDroot**\Emulation\DOC on the M2000 CD).



4.2.5.2 Timer

Promea **timer**

```
DEVICE = ioadr,ZIG1[,"pathname"];
DEVICE = ioadr,ZIG2;
DEVICE = ioadr,ZIG3;
```

The parameters ZIG1/ZIG2/ZIG3 are used to define a Promea timer. If the parameter is not specified, IO address 8010 is assumed.

`ioadr` IO address according to ORG generation (4-digit hexadecimal number).

`pathname` If specified, the emulation stores the relative time (time relative to the Windows time) in the file *pathname* as soon as the time of day is set in ORG-M.. If pathname is not specified, the relative time is lost as soon as the emulation is closed. The emulation then uses the Windows time upon restart.
NT_TIME: if specified, the Windows date/time is set every time the ORG date/time is set.
NT_TIMEMS: if specified, two successive scans do not result in time information with the same *ms* value (due to time stamp in DVS).
If SYS_TIME is specified, ORG interprets the date/time as absolute UTC time (coordinated world time, formerly referred to as GMT), independent of the time zone and summer/winter time.

ZIG1: timer with Windows time resolution accuracy 1) (10ms/15ms). Time balancing acc. to additional parameters ("*pathname*").

ZIG2: timer with millisecond accuracy for time of day reading. Parameters, relative time etc. are ignored.
ZIG2 works implicitly with NT_TIME. The accuracy of the relative time and cyclic clock corresponds to the accuracy of the Windows time resolution.

Problem *Depending on the computer hardware used (processor, clock generator, etc.), a "time running off" effect may be experienced".*

ZIG3: similar to ZIG1 with (implicit) NT_TIME but with millisecond accuracy 1) (Windows time basis of one millisecond accuracy).

Examples

```
DEVICE=8010,ZIG1,"orgzeit";
DEVICE=8050,ZIG1,"$NT_TIME$";
DEVICE=8050,ZIG1,"$NT_TIMEMS$";
DEVICE=8050,ZIG1,"$SYS_TIME$";
DEVICE=8010,ZIG2;
DEVICE=8010,ZIG3;
```

Note When ORG reads the date/time, the emulator reads the Windows date/time. If the date/time is changed via ORG, the emulator changes the Windows date/time only if the pathname NT_TIME or NT_TIMEMS is specified. If ORG date/time and Windows date/time shall run synchronously, pathname NT_TIME or NT_TIMEMS must be specified. A time of day adjustment (e.g., from summer to winter time) must always be initiated by ORG (serial radio clock to PROMEA, manual time adjustment or via *TimeSync.dll*).
□ Automatic summer/winter time change must always be deselected.

1) Accuracy of cyclic and relative time clock and time of day reading.

4.2.5.3 Profibus & Pseudo Devices

PE F7 process element (Profibus)

`DEVICE = ioadr, ALEM, "PROFIBUS", mode, polltime`

This parameter defines a PE F7 device emulation via a **Profibus** interface connection.

ioadr IO address according to ORG generation (4-digit hexadecimal number) , i.a. 2000H

mode Mode, hexadecimal; '0' only is used at present

polltime Cycle time in milliseconds at which alarm generating (input) signal converters are scanned for changes.

Important! The PE F7 emulation requires additional entries

psf =.....

for the routing of the process interface modules in the configuration file *mpar.sys* (see section PE F7 Emulation).

Pseudo device for **PSD**

`DEVICE = ioadr,FTNT,"pipename",length[,vz[,br[,timeout]]]`

The FTNT parameter defines a pseudo device for PSD communication (pipe to SICOMP data). In the SICOMP M system, a visual display unit DSSK must be generated to implement this function. M2000 (*sic_ftnt.exe*) creates a *named pipe* in Windows for every processing direction and takes over the function of a pipe **server**. The data reading pipe receives the ending **`in`**. The data writing pipe receives the ending **`out`**.

ioadr IO address according to ORG generation (4-digit hexadecimal number).

pipename "tnt0" to "ftnt9" are the available pipe names for the use with *m2psapi.dll* (see also PDS documentation). M2000 crates two pipes for every pseudo device - one with the ending **`in`** and one with the ending **`out`** - so that a pipe for each direction is available. If you work without PSD calls, pipe number assignment is not restricted.

length Pipe length in kilobytes. The entry 0 is replaced by the default value 4 kB.

vz Output delay in milliseconds.

br Not specified or ' 0 ': the pipes are not closed at basic parameterization.
Specified and different to ' 0 ': the pipes are closed at basic parameterization.

timeout Timeout for "Call Input monitoring" in ms

Note If `timeout` is set, `vz` and `br` must be specified.

The following applies to the FTNTA (pseudo) device: indicators are returned if a read call is received and data are not available.

Pseudo device for **PSD** **`DEVICE`** `= ioadr,`FTNTX,"*pipename*",*length*[,*vz*[,*br*[,*timeout*]]]

This parameter defines a pseudo device for PDS communication (pipe to SICOMP data). In the SICOMP M system, a visual display unit DSSK must be generated to implement this function. M2000 (*sic_ftnt.exe*) makes use of an existing *named pipe* in Windows and takes over the function of a pipe **client**. The data reading pipe receives the ending **`out`**. The data writing pipe receives the ending **`in.`**

`ioadr`	as above.
`pipename`	as above; the remote computer is specified, a pipe is not created.
`length`	as above
`vz`	as above
`br`	as above
timeout	Timeout for "Call Input monitoring" in ms

Important! The default timeout for "Call Input monitoring" of **10** seconds will be overwritten with the value defined for *timeout*.

Note The following applies to the FTNTAX (pseudo) device:
indicators are returned if a read call is received and data are not available.

with **HardWare(Timer)** **`DEVICE`** `=` *`ioadr,`*`FTNTHW,`"*`pipename`*"`,` *`length`*`[,`*`vz`*`[,`*`br`*`[,`*`timeout`*`]]]`
`DEVICE` `=` *`ioadr,`*`FTNTXHW,`"*`pipename`*"`,` *`length`*`[,`*`vz`*`[,`*`br`*`[,`*`timeout`*`]]]`

All parameters similar as above.

The default timeout for "Call Input monitoring" of **120** seconds will be overwritten with the value defined for *timeout*.

HW in the device parameter stands for HardWare (monitoring timer).

Pseudo device as **pipe server/client**

```
DEVICE = ioadr,FTNTD,"pipename",length[,vz[,br[,timeout]]]
DEVICE = ioadr,FTNTDX,"pipename",length[,vz[,br[,timeout]]]
```

FTNTD creates a pair of pipes and processes the associated pseudo device as a **pipe server**.
FTNTDX establishes connections to a pair of pipes and processes the associated pseudo device as a **pipe client.**
A length word preceding the actual data is not expected by either of the two devices. The devices add a preceding length word to the user data when transmitting and remove it when receiving (see notes). All parameters similar as above.

Example

```
DEVICE=8058,FTNT,"ftnt0",0,700,1,1000   ;DSSK8 pseudo device for PSD;
DEVICE=A308:1,FTNT,"ftnt0",0,0,0,1000   ;DSSK9 pseudo device for PSD;
DEVICE=8028,FTNTX,"\\server1\pipe\ftnt0",0
                                      ;DSSK3 pseudo device for  PSD;
DEVICE=8058,FTNTD,"ftntd0",0          ;DSSK8 pseudo device for  PSD;
DEVICE=8028,FTNTDX,"\\server1\pipe\ftntd0",0;
                                      ;DSSK3 pseudo device for  PSD;
```

Important!

The default timeout for "Call Input monitoring" of **10** seconds
will be overwritten with the value defined for *timeout*.

Note

If `timeout` is set, `vz` and `br` must be specified.

The following applies to the FTNTDA or FTNTDAX (pseudo) device:
indicators are returned if a read call is received and data are not available.

with **HardWare(Timer)**

```
DEVICE=ioadr,FTNTDHW,"pipename",length[,vz[,br[,timeout]]]
DEVICE=ioadr,FTNTDXHW,"pipename",length[,vz[,br[,timeout]]]
```

All parameters similar as above.

The default timeout for "Call Input monitoring" of **120** seconds
will be overwritten with the value defined for *timeout*.

Notes

In ORG-M, the DSS connections used as pseudo devices should be generated with automatic minus acknowledgement. Changes in the generated system can be carried out with TESTSM - however, this should only be attempted by the experienced user.

'Remote - Pipes' allows two SICOMP systems to exchange data directly with each other in M2000. FTNT is set up on computer 1 and FTNTX on computer 2. This way, data can be read out with ‘’STAUAL pseudo device’’ on computer 1, and read in with ‘’STEIAL pseudo device’’ on computer 2 (and vise versa). The Windows function Remote - Pipe handles the transmission over the computer network.

To enable the SICOMP software to communicate with pseudo devices, PSD provides the two pseudo device parameters FTNTD and FTNTDX. The convention ‘’length word" (see next page) can be ignored for these parameters (see above).

Users not working with the standard program *PSDXXX* must observe the following programming convention in Windows, ORG-M, ORG-PV or ORG-HV: a word (SICOMP format) must precede the actual data identifying the data's overall length in words (length word). This convention results from the telegram interface between *PSDXXX* and PSAPI.

Pseudo device for **WINCC-PSD**

DEVICE = *ioadr,*WINCC*,*"*pipename",length*[*,vz*[*,br*[*,timeout*]]]

The WINCC parameter defines a pseudo device for WINCC-PSD. It works similar to FTNT but uses different time stages. In the SICOMP M system, a visual display unit DSSE/DSSA must be generated to implement this function. M2000 creates a *named pipe* in Windows for every processing direction and takes over the function of a pipe **server**. The data reading pipe receives the ending **in**. The data writing pipe receives the ending **out**.

ioadr IO address according to ORG generation
(4-digit hexadecimal number)

pipename Any pipe name can be specified. M2000 creates two pipes for every pseudo device - one with the ending **in** and one with the ending **out** - so that a pipe for each direction is available.

length Pipe length in kilobytes. The entry 0 is replaced by the default value 4 kB.

vz Output delay in milliseconds.

br Not specified or ' 0 ': operating mode is maintained.
Specified and different to ' 0 ': the pipes are closed at basic parameterization.

timeout Timeout for "Call Input monitoring" in ms

Example DEVICE=8058,WINCC,"wincc0",0,100,,2000 ; pseudo device for PSD

Note As soon as *PSDXXX* or *WCCXXX* receive a write instruction for the symbolic CB name "@@@@@@" or the object number "9999", the user subroutine *UPAWP* is called with the associated parameters "address" and "value", which must be linked to *PSDXXX* or *WCCXXX*. Write instructions must always be transmitted for individual words. Read instructions are not permissible.

For users of WinCC and WinCC-PSD channel DLL, this means that an external PSD variable of type 'A' (unsigned 16-bit value) can be defined and assigned to the CB name "@@@@@@" and the address "0"(=V128). If the value "100" is written to this variable (WinCC function "SetTagWord"), the *UPAWP* subroutine is called in ORG-M or ORG-PV with G6:=0 and G7:=100. This allows users to activate any (self-programmed) function from within WinCC under the emulated ORG. The criteria used are the V address in "pseudo CB" 9999 and the transmitted value.
PSD variables assigned to CB "@@@@@@" may not be updated by WinCC!
The standard elements *PSDXXX* and *WCCXXX* contain a dummy module. This dummy module with the designation *PLSKx-PSD.PSDAWP* is supplied on the virtual disk PLSK.PSD and can be used as a template (it simply indicates the values transmitted for G6 and G7 on the standard signaling device).

Example

ORG shall be booted with and without start list via WinCC buttons.
CB "@@@@@@" and address "3" (=V130) are assigned to the PDS variable "ORGBOOT" (unsigned 16-bit value).
One button sets the value 1 (SetTagWord (ORGBOOT,1)), another button the value 2 (SetTagWord (ORGBOOT,2)). *UPAWP* first checks whether the "Boot ORG" function is meant by comparing G6 with 3. It then verifies via G7 (1 or 2), whether ORG is booted with or without start list and finally executes the function accordingly ($RUFORG"...).

Note

When old linker cards are used with new basic language elements, error message "/NDEF:UPAWP" is displayed upon loading *PSDXXX* and *WCCXXX*. This error message can be ignored as long as the new function is not used.

Dummy module

```
/#PSDAWPXX                    170800/PJG
**********************************************************************
***      PSDXXX:   USER FUNCTION
**********************************************************************
         'PKEN' ,,,,0
         'DEEX' UPAWP
         'NDEX' STAMELD

         'NAME' PSDAWP
         'RECORD' PSDAWP

'VA' TEXTA/
' Y'      = 'Z=PSDAWP: DUMMY: #### ####(3)'
'VA' TEXTE/

'IA' UPAWP/   ('R2) := R7        *** DUMMY!
' A'      ('R2) := R3       ***
' A'      ('R2) := R4       ***
' A'      ('R2) := R5       ***

' A'      *** G6          *** RELATIVE ADDRESS (0 and above)
' A'      *** G7          *** CURRENT VALUE

' A'      R3 := <7+TEXTA>          ***
' A'      R4 := 'H=8000'           ***
' A'      R5 := G6                 ***
' A'      $BIHEX;                  ***              N
' A'      'IA'                                      B
         ='H=E000'                           B
          ***                                       B

' A'      R3 := <10+TEXTA>         ***
' A'      R4 := 'H=0000'           ***
' A'      R5 := G7                 ***
' A'      $BIHEX;                  ***              N
' A'      'IA'                                      B
         ='H=E000'                           B
          ***                                       B

' A'      R5 := (R2')       ***
' A'      R4 := (R2')       ***
' A'      R3 := (R2')       ***

' A'      :SP (R2')         ***

         'END'
```

4.2.5.4 Promea MX & BDE Terminal (ES 1)

Promea-MX

DEVICE = *ioadr*,EAMX,"*pathname*" [,*flags*,*verz1*,*verz2,verz3*]

This parameter defines the emulation of a PROMEA MX via any serial interface. The serial interface must be defined as COMxx in Windows.

ioadr IO address according to ORG generation (4-digit hexadecimal number).

EAMX Device designation

pathname Pathname of the corresponding serial interface in Windows (COMx or \\.\COMxx).
For EAMX, the serial interface can also be on a remote computer. The pathname is specified in the form REM:x. Parameter x denotes the port number of the first of two TCP/IP sockets created by the device process on the M2000 computer.
(see section Interface Concentrator)

flags pabc (hexa parameters)

a=0	No DEVICE INOPERABLE detection
a=1	DEVICE INOPERABLE detection for TTY interface (jumper between RxD and CTS required)
a=2	DSR scan prior to output to V.24 interface
b=0	Interface may be overparameterized
b=1	Protect interface from overparameterization
b=2	Protect interface from resetting (octs =... and odsr =... of *mode* command effective)
b=3	Protect interface from resetting and overparameterization
c=0	DISIT PB is preset
c=1	GS01 PB is preset
c=2	Connection 'old DISIT devices' (no XON/XOFF)
p=0	Buffer size 10 kB
p=1	Buffer size 128 kB

verz1 Delay period in ms before data are transmitted (if the connected device is too slow)
verz2 Delay period in ms after data are transmitted
verz3 Delay period in ms before data are received

Note This operating mode (*flag* c=2) is required if DISIT devices originally connected to R-systems shall now be operated in the M-emulation.

Important ! Printers generated in the M-system as MX devices should also be defined as EAMX devices in the MPAR.SYS file. Otherwise error messages such as
'ERROR: Number of interrupt vectors'
'ERROR: DEVPRB, SWCTR unknown'
are issued.

Examples PROMEA MX interface on a local COM interface

```
DEVICE=8202,EAMX,"\\.\COM2",1101,000,000;
DEVICE=A302:1,EAMX,"\\.\COM128",0012,000,000;
```

PROMEA MX interface on a remote Windows system

```
DEVICE=8202,EAMX,"REM:24000",101,000,000;
```

MSHUP

`DEVICE` = *`ioadr`*`,MSHUP,"`*`pfadname`*`"[,`*`flags`*`];`

This parameter defines the emulation of a special module (MSHUP) for PROMEA MX via any serial interface. The serial interface must be defined as COMxx in Windows.
MSHUP (*sic_mshp.exe*) emulates the special module for Promea MX on triggering the horn in MADAM-S systems.

`ioadr` IO address according to ORG generation
(4-digit hexadecimal number).

MSHUP Device designation

`pfadname` Pathname of the corresponding serial interface in Windows
(COMx or \\.\COMxx).

Hinweis The output is carried out via the RTS line of the serial interface.

Status of RTS signal:

flags (parameterisation hexadecimal)

		passive state	output impulse
0000		off	on
8000	output inverted	on	off

Examples Special module on a local COM interface

`DEVICE=8040,MSHUP,"\\.\COM7";` no *flags* or 0000

`DEVICE=8048,MSHUP,"COM2",8000;` output inverted

ES **DEVICE** = *ioadr*,EAES,"*pathname*"[,*flags*[,*test*]]

This parameter defines the connection of a BDE terminal of the ES 1 device family via serial interface.

ioadr IO address according to ORG generation
(4-digit hexadecimal number).

EAES Device designation

pathname \\.\COMx for serial standard interface

flags Parameters (Hexa)
Bits 0 - 3:Serial interface parameters
=0: Define the interface with the 'mode' command
before starting the emulation
=1: Transfer rate: 1200
(terminal basic setting)
Data bits: 7
Stop bits: 2
Parity: even
=2: Transfer rate: 2400
Data bits: 7
Stop bits: 2
Parity: even
=3: Transfer rate: 4800
Data bits: 7
Stop bits: 2
Parity: even
=4: Transfer rate: 9600
(Hella setting)
Data bits: 7
Stop bits: 2
Parity: even

Bit 4: Input mode
=0: Input not transparent
=1: Input transparent

Bit 5: Output mode
=0: Output not transparent
=1: Output transparent

test For testing only
=0: (no test) normal mode
=1: Output baud rate and input/output mode to
‘Parameterize’

Example

```
DEVICE=8202,EAES,"\\.\COM2",34;
DEVICE=A302:1,EAES,"\\.\COM188",34;
```

(Input transparent and output transparent (3), 9600 bauds (4))

5 Configuration File for SIC-R

5.1 General Entries

CPU type

`CPU` = *`typ`*

The specification of the CPU type (*typ)* is particularly important for machine intimate programs and the accuracy of calculation results.

The following CPU types are supported: R10, R10V, R20, R30

Example

```
cpu = r10v;
```

Main memory size

`MAXMEM` = *`size`*

Size of the main memory denoted as decimal number in the range from 64 to 1024 (in KW). This corresponds to the maximum size of 2 MB of the original system. The parameter can be taken from the generation log of the original system.

Example

```
maxmem = 512;
```

Main memory retrieval file

`IMAGE` = "*`pathname"[,mode]`*

This parameter defines a Windows file to which M2000 writes the main memory contents of the emulated SICOMP R (emulation of a battery-backed main memory) upon normal termination of the program (`Stop` button on `Monitor` tab of M2000).

`pathname` Complete path of the Windows file. The file is set up implicitly.

`mode` Processing mode of the image file.

NEW (default). The image file is generated anew under the same name with every termination of the program. An existing image file will be deleted.

ALTx The parameter ALTx (x=1 to 9) generates x image files which are alternately overwritten. The image files are designated *filename.im1....filename.imx*. The header of every image file contains the date/time specification and a consistency identifier.

At the start of the write process, an inconsistency identifier is entered. After the HSP contents has been completely stored, a consistency identifier with time/date specification is set. The oldest or an inconsistent file will be overwritten.

When the emulation is restarted, the main memory of the emulated SICOMP M systems is pre-allocated from this main memory retrieval file. The corresponding identifiers for battery-backed restart are transmitted to the ORG system and the DVS files can be reconstructed with DFCONS.

Example

```
IMAGE = "\user\pcsic\sysrett.v1";
```

Bootstrap loader **`BOOT`** = *`hexboot`*

hexboot is a 4-digit hexadecimal that specifies the bootstrap loader. This number corresponds to the switch settings on the maintenance panel.

Example

```
BOOT = 0025;
```

Deactivate floating-point processor **`NOFPP`**

The floating-point processor performs the arithmetic operations for 32-bit fixed-point and 32-bit and 64-bit floating-point operations and is activated in the emulation by default. You can use this instruction to deactivate the function of the floating-point processor for specific applications.
(NOFPP = No Floating Point Processor)

Example

```
NOFPP
```

MEC-28R **`MECR`** = *`anr`*

This parameter is required for the operation of an MEC-28R interface connection (must be set for each individual interface connection).

`anr` Connection point number of the interface connection (decimal number)

Example

```
MECR = 7;
```

5.1.1 Visual Display Units

Visual display units, serial connection

DEVICE = *anr,gnr,keyword,*"*pathname*"[,*upar*[,*kennung*[,*verz*]]]

This parameter defines the connection of a visual display unit via serial interface (COMxx).

anr,gnr Connection point number/device number according to ORG generation (decimal number)

keyword The following visual display units can be specified:

3974	Connection to C71458-A6404-A interfacing
3974R	Connection to PROMEA module
3974M	Connection to PROMEA module

The *keyword* parameter must be assigned as followed:

3974R	for TE2000 AX or TE-ALPHA
3974M	for TE2000 DX or TE-DISIT

pathname Designates the serial interface defined in Windows. The *pathname* is denoted as COMx or \\.\COMxx.

upar Mode 1: Overparameterization (hexa)
= 0: Overparameterization possible
= 1: Protect interface from overparameterization (default setting)
Mode 2: Device inoperable detection (hexa)
= 100: Device inoperable detection for TTY interface
(jumper between RxD and CTS required)
= 200: DSR scan prior to output to V.24 interface
Modes 1 and 2 can be linked by an OR operation.

kennung Enables external/internal dongle
IPKS identification (kennung): Enables the (old) functions of the TE-ALPHA/TE-DISIT terminal emulation in the central M2000 dongle.
TE2000 identification (kennung): Enables the functions of the TE2000AX/DX terminal emulation in the central M2000 dongle.
NONE identification (kennung): Function enabling in external dongle required.

verz Delay period (milliseconds) in 32-bit representation.
The high-order 16 bits are interpreted as input delay, the low-order bits as output delay.

Notes

- The interface must be defined accordingly with the Windows *mode* command.
- Flow control is set to XON/XOFF=93/94h by M2000.
- See table 1 in this section (visual display unit types).
- Commas must be entered in the parameter line even if no options are selected.

Example

```
DEVICE=1,0,3974R,"\\.\COM3",100,IPKS,0x00640032;
```

Terminal connection (TTY) with device inoperable detection, central function enabling, input delay (Verz)=100ms, output delay (Verz)=50ms.

```
DEVICE=1,1,3974M,"\\.\COM4",1,NONE,0x00640032;
```

DISIT version, overparameterization not possible, external dongle, input delay (Verz)=100ms, output delay (Verz).=50ms.

Terminal emulation, **local**

DEVICE = *anr,gnr,keyword,"pipename"*

This parameter activates a terminal emulation as local Windows application.

anr,gnr Connection point number/device number according to ORG generation (decimal number)

keyword The following type specifications are permitted:
3974RT for a local console
3974MT for a local DISIT emulation

pipename Unique name of the console application. A *named pipe* is created under this name, which must unambiguous throughout the system. This name is also displayed in the window heading.

Notes

- If used in a window, the screen resolution should not be set below 800x600 pixels (16/256 colors)
- This parameter can be defined several times to emulate several display units. In this case, each device is displayed in a separate window.
- A powerful graphics card (PCI) should be used to enhance performance.
- The device emulation's scope of performance is briefly described in section 6.1 Device Emulation DS074. A detailed description is given in the User Manual **TE2000 Terminal Emulation**.
- Do not include special characters in the pipe name, such as (- \ , / . ?, etc.) to avoid that several terminal emulations work with the same pipe.
  Example: Instead of **A-DSSK002**, write **A_DSSK002**

Example

```
DEVICE=1,1,3974RT,"DSSE1";
```

DISIT emulation
TE2000 DX
via network

DEVICE = *anr,gnr,*3974MNET,"*socketnr[/t]*"*[,*TE2000*[,timer]]*

This parameter defines the operation of a DISIT terminal emulation TE2000 DX via network.

DISIT emulation
TE-DISIT
via network

DEVICE = *anr,gnr,*3974MNET*,* "*socketnr[/t]*"*[,*IPKS*[,timer]]*

This parameter defines the operation of an (old) TE-DISIT terminal emulation via network.

anr,gnr Connection point number/ device number according to ORG generation (decimal number).

socketnr Socket number for the communication between the M2000 system and the terminal emulation. The socket number must be transmitted as start parameter when the remote terminal emulation is started. Socket port number according to administration
Recommended range: >10000
Parameter ***/*** **t**, see next page.

TE2000 Specifies that this terminal emulation is enabled in the central M2000 dongle.

IPKS Specifies that this terminal emulation is enabled in the central M2000 dongle .

timer Time in ms the inputs/outputs are delayed to accommodate slow visual display units (e.g., ES05) and user programs.

Examples

```
DEVICE=1,2,3974MNET,"20001";
DEVICE=2,1,3974MNET,"20011/30",TE2000,1500;
```

Terminal emulation
TE2000 AX
via network

DEVICE = *anr,gnr*,3974RNET,"*socketnr[/t]*"*[*,TE2000*[*,*timer]]*

This parameter defines the operation of a TE2000 AX terminal emulation via network.

Terminal emulation
TE-ALPHA
via network

DEVICE = *anr,gnr*,3974RNET,"*socketnr[/t]*"*[*,IPKS*[*,*timer]]*

This parameter defines the operation of an (old) TE-ALPHA terminal emulation via network.

anr,gnr Connection point number/device number according ORG generation (decimal number).

socketnr Socket number for the communication between the M2000 system and the terminal emulation. The socket number must be transmitted as start parameter when the remote terminal emulation is started.
Socket port number according to administration
Recommended range: >10000
The established connection between M2000 and the terminal emulation will be time-monitored. If no data are exchanged within **t** seconds (default value: 180s), M2000 terminates the connection. The monitoring period ***t*** can be parameterized with the socket number (see examples).

TE2000 Specifies that this terminal emulation is enabled in the central M2000 dongle (TE2000AX/DX).

IPKS Specifies that this terminal emulation is enabled in the central M2000 dongle (TE-ALPHA/TE-DISIT).

timer Time in ms the inputs/outputs are delayed to accommodate slow visual display units (e.g., ES05) and user programs.

Examples

```
DEVICE=1,3,3974RNET,"20011";
DEVICE=2,1,3974RNET,"20011/30",TE2000,1500;
```

Table 2 Display unit types and keywords for *rpar.sys* parameter definition

Display unit	**Keyword**		
Type	Serial connection	Local terminal	Network terminal
3974	3974	—	—
3974R	3974R	—	—
3974M	3974M	—	—
DS074	3974R	—	—
DS075	3974R	—	—
DS075 DISIT	3974M	—	—
TE2000 AX	3974R	3974RT	3974RNET
TE2000 DX	3974M	3974MT	3974MNET
TE-KONS	—	3974RT	—

Note If two or more network cards are installed in the computer, the IP address (or surrogate name) of the corresponding card can be assigned to a specific visual display unit. The surrogate name (alias) must be entered in the *hosts* file (**systemdir***system32\drivers\etc\...*). The corresponding parameter line is structured as follows:

DEVICE = *anr,gnr,*3974RNET, *ipadr*:*port,kennung*
DEVICE = *anr,gnr,*3974MNET,*aliasname:port,kennung*

Example

```
DEVICE=2,0,3974RNET,192.168.2.10:20010,IPKS;
```

```
DEVICE=2,0,3974RNET,NetCard-LAN1:20010,IPKS;
```
The connection is established via network card 1.

```
DEVICE=2,1,3974RNET,192.168.2.13:20013,TE2000;
```

```
DEVICE=2,1,3974MNET,192.168.2.13:20013,IPKS;
```

```
DEVICE=2,1,3974MNET,NetCard-LAN2,20013,TE2000;
```
The connection is established via network card 2.

Note A detailed description is given in the User Manual:
TE2000 Terminal Emulation.

5.1.2 Storage Media

Disk drive occupation mode

mode = share

MODE specifies the occupation mode (exclusive/shared) of the disk drives.
If the mode parameter is not set, `exclusive` mode will be active (default setting). The use of the MODE parameter allows you to access the disks from both system environments.

Disk drive **3941,3942,3945,3946, 3947,3948,3949**

DEVICE = *anr,gnr,keyword,*"*pathname*"[*,flags*]

This parameter is used to define the emulation of a disk drive by means of a Windows file.

anr Connection point number of the device according to ORG generation (decimal number).

gnr Device number according to ORG generation (decimal number).

keyword

3941	designates a device with 9744 sectors
3942	designates a device with 48720 sectors
3945A	designates a device with 2048 sectors
3945B	designates a device with 4096 sectors
3945C	designates a device with 8192 sectors
3946	designates a device with 8192 sectors
3947	designates a device with 15616 sectors
3947A	designates a device with 15616 sectors
3947H	designates a device with 1034240 sectors
3948	designates a device with 64640 sectors
3948A	designates a device with 64640 sectors
3948B	designates a device with 61408 sectors
3949	designates a device with 25856 sectors
3949A	designates a device with 25856 sectors
3949B	designates a device with 25856 sectors
3949C	designates a device with 25856 sectors

pathname Location of the corresponding Windows file (the complete path must be entered). If the file does not exist, it is created by the emulator in the defined length. The file can then be formatted and set up with INIT.

flags Definable parameter (hexa)
= 1 WRITE THROUGH
This parameter ensures that changed data is written to the disk and doesn't remain in the CACHE. This is a useful option for DVS data carriers.

Example

```
DEVICE=5,2,3948, "c:\sicomp\platten\SYSR_PS2.sic",1;
```

Floppy disk drive

DEVICE = *anr,gnr,keyword,"pathname"*

This parameter is used to define a floppy disk drive.

anr Connection point number according to ORG generation (decimal number).

gnr Device number according to ORG generation (decimal number).

keyword

3943	designates a device with 256256 bytes
3944	designates a device with 1021696 bytes
FLOP	designates a device with 1228800 bytes

Operation of a floppy disk drive (3.5" 1.44MB or 5.25" 1.2MB) without compatibility to SICOMP M. This floppy disk drive can only be used for data backup and the data exchange between several emulations. In the emulated system, the floppy disk drive can be used like a 3943/3944 drive (also in connection with INIT). However, a diskette written with FLOP cannot be read on a SICOMP M computer.

pathname The pathname is: \\.\A:
\\.\B:

Example

```
DEVICE=3,0,3944,\\.\A:;
```

Extended **floppy disk access**

M2000 is capable of processing DOS formatted floppy disk files that are compatible with the following SICOMP R data media:

- 8 inch floppy disk replacement based on EDS system technology, with 908 PC card and BDK program.

This floppy disk access is set up in the parameter file *rpar.sys* as follows:

Pathname: for example "A:\EKTEST" instead of \\.\A: only.
M2000 looks for the specified file (here EKTEST) on the floppy disk. If the file exists on the floppy disk, M2000 will process this file. If the specified file (here EKTEST) does not exist, M2000 looks for a file of appropriate length and uses it if it finds one. If a file of appropriate length is not found, M2000 creates a file under the specified name (here EKTEST) on the floppy disk. Thus the name specified in the parameter file (here EKTEST) is only relevant if M2000 has to create a new file. Existing files can use any name.

Magnetic tape cassette - drive **MK82**

DEVICE = *anr,gnr,*MK82*,"pathname"*

This parameter allows you to operate an MK82 compatible magnetic cassette drive of the type Tandberg Data Panther TDC 3660/3820 for writing and reading 150 MB cassettes.
Magnetic tape cassettes of an MK80/81 device with 60MB can be read only.
This parameter requires the definition of an MECR instruction.

anr Connection point number of the device (decimal number).

gnr Not relevant

pathname Pathname of the corresponding Windows device

Example

```
DEVICE=7,0,MK82,\\.\Tape0;
MECR=7;
```

Optical disk OP11

DEVICE = *anr,gnr,*OP11*,"pathname",scsi-id*

This parameter defines the emulation of a rewriteable OP11 optical disk via the SONY RMO-S580 drive.
This parameter requires the definition of an MECR instruction.

anr Connection point number of the device (decimal number).

gnr Device number (decimal number).

pathname The pathname is \\.\PhysicalDrive%.
% is the disk number in Windows, starting with 0 (see Windows hard disk manager).

scsi-id SCSI drive number (decimal number, 0..7) of the emulated drive in the original system, not in the PC.

Example

```
DEVICE=7,0,OP11,"\\.\PhysicalDrive1",3;
MECR=7;
```

Removable magnetic disk **WP256**

DEVICE = *anr,gnr,*WP256*,"pathname",scsi-id*

WP256 defines the emulation of a removable magnetic disk drive (e.g., SYQUEST). This parameter requires the definition of an MECR instruction.

anr Connection point number of the device (decimal number).

gnr Device number (decimal number)

pfadname The pathname is \\.\PhysicalDrive%.
% is the disk number in Windows, starting with 0.

scsi-id SCSI drive number (decimal number, 0..7) of the emulated drive in the original system, not in the PC.

Example

```
DEVICE=7,0,WP256,"\\.\PhysicalDrive1",1;
MECR=7;
```

5.1.3 Printers

Printer operation, parallel port

`DEVICE` = *`anr,gnr,`*`DRUA`*`,"pathname"[,flags[,timer]]`*

This parameter diverts the *DRUA* device to the printer connected to the parallel interface.

`anr` Connection point number of the device according to ORG generation (decimal number).

`gnr` Device number according to ORG generation (decimal number).

`pathname` Pathname of the corresponding Windows device, usually LPT1.

`flags` Identification:

3915	Printer 3915 (ES902 or PROMEA)
3915B	Printer 3915 (SIVAREP-B)
3916	Printer 3916 (PROMEA)
3916B	Printer 3916 (SIVAREP-B)
3916E	Printer 3916 (ES902)
3918	Printer 3918
3919	Printer 3919
CRLF	Replacement LF with CR/LF at transfer end

`timer` WRITE monitoring period in seconds (default value: 20 s)

Example

```
DEVICE=1,1,DRUA,"LPT1";
```

Important! No printer driver should be installed or activated for the parallel interface in Windows.

Printer output via **Windows printer driver**

`DEVICE` = *`anr,gnr,`*`DRSPOOL`*`,"pathname"`*`[,`*`flags`*`[,`*`timer`*`]]`
`DEVICE` = *`anr,gnr,`*`DRSPOOLF,"`*`pathname`*`"[,`*`flags`*`[,`*`timer`*`]]`

This parameter is used to pass on the DRUA device to a Windows printer driver. The first output opens a document, which is closed after a default period of 5 seconds if no subsequent outputs follow. A new document is opened after the default period of 5 seconds has elapsed.
DRSPOOLF furnishes the outputs with the trailing control character FF (FormFeed) required by some laser printers.

The parameters correspond to those for parallel output (DRUA parallel, see above).

Example

```
DEVICE=1,1,DRSPOOL,"LPT1";
DEVICE=2,1,DRSPOOL,"\\DRSERVER\DRUA00",10;
DEVICE=2,1,DRSPOOLF,\\192.168.1.7\DRUA10;
```

Note The following applies to network printer outputs:

`DRSERVER`	= Name of the printer server
`192.168.1.7`	= IP address of the printer server
`DRUA00`	= Local enable name of the network printer
`DRUA10`	= Local enable name of the network printer

Printer output to **Windows file**

DEVICE = *anr,gnr,*DRUA*,"pathname"[,flags]*

This parameter is used to transfer the printer output to a Windows file. The output starts at the beginning of the file. An existing file will be deleted.

anr Connection number of the device according to ORG generation (decimal number).

gnr Device number according to ORG generation (decimal number).

pathname Complete path of the windows file to which the output is transferred. The file is set up implicitly.

flags Identifier (see DRUA)

Example

```
DEVICE=1,1,DRUA,"drua0";
```

Printer operation (serial) with **DR202**

DEVICE = *anr,gnr,*DR202*,"pathname"[,flags[,timer]]*

This parameter allows you to operate a DR202 printer connected to the serial interface.

anr Connection number of the device according to ORG generation (decimal number).

gnr Device number according to ORG generation (decimal number).

pathname Pathname of the corresponding serial interface in Windows.

flags Definable parameter (hexa)
= 1 BREAK detection

timer WRITE monitoring period in seconds (default value= 20 s)

Example

```
DEVICE=1,1,DR202,"COM7";
```

Printer operation (serial)
all printers

`DEVICE` = *`anr,gnr,`*`DRCOM`*`,"pathname"[,typflags[,time1][,time2]]`*

This parameter allows you to operate all printers connected to the serial interface.

`anr` Connection number of the device according to ORG generation (decimal number).

`gnr` Device number according to ORG generation (decimal number).

`pathname` Pathname of the corresponding serial interface in Windows.

`typflags` Parameters (hexa):
printer type ''typ'',
string ''CRLF'',
or flags:
= 1 Break detection for TTY
= 2 Break detection for V.24
= 4 Ready/busy
= 8 Do not define serial port parameters - the MODE command applies
= 10 Similar to parameter(string) CRLF

`time1` WRITE monitoring period (default value = 20 s).

`time2` Time delay in ms before data transmission is started (if the connected device is too slow).

Example

```
DEVICE=1,1,DRCOM,"COM7",3915,,100;
```

Printer output to
Windows file
(parameter-controlled)

DEVICE = *anr,gnr*,DRSPOTX,"*file*"

DRSPOTX defines the parameter-controlled printer output to a Windows file.

anr Connection point number of the device according to ORG generation (decimal number).
gnr Device number according to ORG generation (decimal number).
file (Path)name of the parameter file.

Note

The following rules must be observed when a parameter file is created:
Comments start with a semicolon.
The parameter names SUPPRESS, CHANGE, START, STOP may be used up to 255 times, a string can have a maximum length of 64 charcaters, all characters can be used.
All other parameter names may only be used once (if a name is used more than once, always the last definition applies).
In character strings, special characters can be represented as hexa patterns (see below):
The two characters following a backslash (\ ) are interpreted as hexa pattern. The \ character itself can only be represented as \5d.
The **"** character can only be represented as \22.
Only one START or STOP string will be interpreted in every output record.
If the parameter file is changed during running operation, the changes will become effective after the emulator has been restarted, or after the "Reset device" function has been activated by the TWR.EXE application (open files will be closed before the reset function is executed).
Time-controlled one-time or reoccurring processes must be implemented with the Windows command **at** *time command.* This requires that Windows SCHEDULE+ is started (**net start** schedule).

Example

at 23:00 netcopy.bat

Dynamic functions are processed in the following sequence:
1. SUPPRESS: all characters to be deleted are removed.
2. CHANGE: the necessary replacements after deletion are carried out.
3. START: checks whether the data set contains a 'start character sequence' after deletion and replacement.
4. STOP: checks whether the data set contains a 'stop character sequence' after deletion and replacement.

Parameter names:

PATH [*drive*:[*path*]]	; Location where the files to be created are stored (default: M2000 start directory)
SPOTTIME *sec*	; Definition of the monitoring time (default: 0, no monitoring time specified).
DOSNAMES	; Use of DOS compatible file names,
NTNAMES	; Use.of file names not compatible with DOS (default).
LFDNAMES	; File names are created in the form of 'consecutive numbers'
STARTNAMES *n*	; *n* characters following START "*string*" form the beginning of the file name (z...z_JJJJMMTThhmmss.ioadr)
EXTENSION x....x	; Freely definable (file) extension
SHARE	; The exclusive allocation status of the output files is canceled.
SUPPRESS "*string*"	; Characters (sequences) are removed prior to output
CHANGE "*string1*" "*string2*"	; Characters (sequences) are replaced prior to output, *string*1/*string2* can have different lengths.

```
START "string"                  ; Definition of a 'start character sequence'
STOP "string"                   ; Definition of a 'stop character sequence'
WINDOW "name"                   ; Outputs also shown in window name
WINDOWT "name"                  ; as above, but with preceding time stamp
COMMAND "[drive:[path]] file"   ; Definition of a command;
```

The command is transmitted after the target file is closed. 5 parameters can be specified:

The 1st parameter designates the currently closed file,
the 2nd parameter contains 'file closing event' identifications such as:

"to"	*Period*
"aa"	*Start character sequence*
"zz"	*Stop character sequence*
"bo"	*Boot ORG* or reset printer
"nt"	after termination of the emulation

The 3rd parameter specifies the directory where the file is stored,
the 4th parameter specifies the name of the currently closed file,
the 5th parameter specifies the extension of the currently closed file.

Example `DEVICE=2,2,DRSPOTX,"\sicomp\spotx.par";`

The parameter file *spotx.par* could look as follows:

```
PATH e:\temp                                                        ; ( 1 )
DOSNAMES                                                            ; ( 2 )
EXTENSION DRUA1                                                     ; ( 3 )
SPOTTIME 20                                                         ; ( 4 )
SUPPRESS "\0D\0A\03"                                                ; ( 5 )
CHANGE "\0D\0A" "\0A"                                               ; ( 6 )
START "//JOB:"                                                      ; ( 7 )
STOP   "//JEND"                                                     ; ( 8 )
STARTNAMES 7                                                        ; ( 9 )
SHARE                                                               ; ( 10 )
COMMAND "e:\temp\drspotx.bat" "drfile1" "to" "\temp" "drfile1" "txt"  ; ( 11 )
```

(1) The print files are stored in the directory *e:\temp*.

(2) DOS compatible file names are used.

(3) The freely definable (file) extension is .DRUA1 instead of *.ioadr .*

(4) The print files are closed if no output is initiated within the period of 20 seconds.

(5) The character sequence CR-LF-ETX is suppressed.

(6) The character sequence CR-LF is replaced by LF.

(7) The current print file is closed as soon as the character sequence "//JOB:" is detected; a new file is created and "//JOB:" is transferred to the new file.

(8) The current print file is closed as soon as the character sequence "//JEND" is detected and the string "//JEND" is written to the end of the file that is closed. Then a new print file is opened to which data are written until the output ends (as defined).

(9) The first 7 characters following "//JOB:" (e.g., JOBCOPY), form the beginning of the new file name, e.g., *JOBCOPY_20020812135801.0202*.

(10) The (file) status *shared allocation* is activated, the file can be viewed during output.

(11) The print file *drfile1.txt* is closed after the period (SPOTTIME) has elapsed (identification *to*). After the file is closed, *drspotx.bat* is activated.

Note If SPOTTIME is set to 0 (default value), the file created last remains open and "receptive" until another event causes the file to close (START ..., STOP ..., etc.).

Printer output to **Windows window**

DEVICE = *anr,gnr,*DRWIN*,"fenster"*,"datei1","datei2","size"

The printer output is transmitted to a Windows window and can also be stored in files.

anr Connection point number of the device according to ORG generation (decimal number).

gnr Device number according to ORG generation (decimal number).

fenster Window name

datei1 First log file

datei2 Second log file

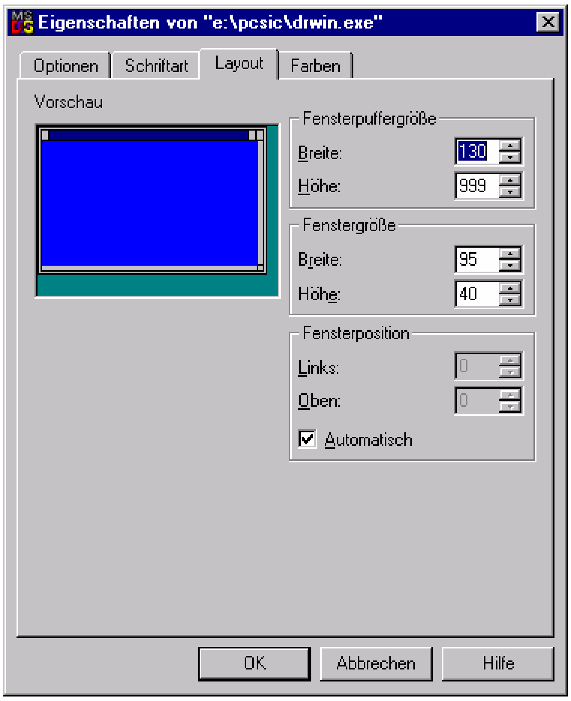
size Max. file size in kilobytes

Example

```
DEVICE=1,1,DRWIN,"DRUA0";
DEVICE=2,2,DRWIN,"DRUA","outfile1","outfile2","10";
```

The DRWIN parameter diverts the printer outputs to a Windows window. After the start of the emulation, the screen displays an additional window named DRUA0. This window can be configured. Right-click the header of the DRUA0 window and select *Properties.* Select *Layout ⇒ Window buffer size* - Window size and specify the number of printer lines the window shall be capable to buffer (e.g., 999). The window now works like a cyclic buffer with a size of 999 lines.

Now, to transmit parts of the printer outputs (for example, error messages) to the computer's printer, proceed as follows:
Right-click the header and select *Edit ⇒ Select*. Select the lines you want to print. Copy the selected area to the clipboard (CTRL+C). Open a text editor, such as WRITE, and paste the selected area into a document (CTRL+V). Then print out the document.



Fensterpuffergröße
Breite: 130
Höhe: 999

Fenstergröße
Breite: 95
Höhe: 40

Fensterposition
Links: 0
Oben: 0
☑ Automatisch

(Additional) output to log files

Additional output to log files can be implemented with the parameters *file1*, *file2* and *size* (see example 1).

The printer output is transferred to *file1* first; if *file1* has reached maximum size (*size**1024), the printer output is transferred to *file2*; if *file2* has reached maximum size, the printer output is stored in *file1* after the old contents has been deleted.

Always a complete printer line is written to the file before the file is scanned for maximum size.

The current settings are stored in the *window.par* (e.g., *DRUA0.par*) when the emulation is terminated.

If the emulation is restarted and the parameters haven't been changed, the printer outputs are added to the log file written last followed by the same procedure as described above.

If the parameters have been changed or if problems are encountered opening the file written last, the printer outputs are transferred to a new *file1*.

A message line (--End DRWIN--) is added at the end of the current log file when the procedure is terminated.

Printer output with date/time to a **Windows window**

DEVICE = *anr*,*gnr*,DRWINT,"*window*"

The DRWINT parameter is similar to the DRWIN parameter; date and time are added at the beginning of each line of the printer output in the following format: `YYYY-MM-DD hh:mm:ss.ms`. This format is ideal for message printers.

anr	Connection point number of the device according to ORG generation (decimal number).
gnr	Device number according to ORG generation (decimal number).
window	Window name

Examples

```
DEVICE=2,2,DRWINT,"DRUA";
```

DRWINT example

```
2002-08-1311:58:02.701 CDLI TO DRUA
2002-08-13 11:58:02.701 TESTCD 28 CA CBUET 65271 42
```

Note

Entering `DRWINMIN/DRWINTMIN` instead of `DRWIN/DRWINT` minimizes the printer windows.

5.1.4 Interface Connections

DUST3964
computer interfacing
via any serial interface

`**DEVICE**=*anr*,gnr,3964,"*pathname*",*p1[,p2[,p3[,p4]]]*`

This parameter defines the emulation of a DUST3964 computer interface connection via any serial interface. The serial interface must be specified as COMxx under Windows.

anr Connection point number of the device according to ORG generation (decimal number).

gnr Device number according to ORG generation (decimal number).

pathname Pathname of the corresponding Windows interface, i.e. COMx or \\.\COMxx. The interface must be predefined with the Windows command *mode* (default setting). Overparameterization through ORG is implemented to a large extent.

p1 Protocol parameters **abcd**:

a	0 = POA 50 used 1 = POA 10 used
b	0 = low priority 1 = high priority
c	0 = ORG timeout = 2s 1 = ORG timeout = 30s
d	0 = without BCC 1 = with BCC

p2 = *value* (ms)	timeout – waiting for ZE character reception (default = 200ms)
p3 = *value* (ms)	Delay period output in ms (default value 0ms)
p4 = *value* (ms)	Delay period input in ms (default value 0ms)

Example

```
DEVICE=4,1,3964,"\\.\COM21",0111,300,100,100;
```

Operation at port 21,

p1 POA 50 / high priority / ORG timeout 30s / with BCC,
p2 timeout 300ms,
p3 delay period output 100ms,
p4 delay period input 100ms,

DUST3964R
computer interfacing
via any serial interface

DEVICE=*anr*,gnr,3964R,"*pathname*",*par[,t_ze[,verzaus[,verzein]]]*

This parameter defines the emulation of a DUST3964R computer interface connection via any serial interface. The serial interface must be specified as COMxx under Windows.

anr Connection point number of the device according to ORG generation (decimal number).

gnr Device number according to ORG generation (decimal number).

pathname Pathname of the corresponding Windows interface, i.e. COMx or \\.\COMxx. The interface must be predefined with the Windows command *mode* (default setting). Overparameterization through ORG is implemented to a large extent.

par Protocol parameters **tpb**:

t	0 = BFA monitoring time = 2 sec 1 = BFA monitoring time = 30 sec
p	0 = low priority 1 = high priority
b	0 = without BCC 1 = with BCC

t_ze Max. ORG response time (in ms) to BFA input
(entry = 0 corresponds to the default value of 200ms)

verzaus Delay period (in ms) for 'BFA output' instruction
(for slow peer systems; default value = 0)

verzein Delay period (in ms) for 'BFA input' instruction
(for fast emulator computers; default value = 0)

Example

```
DEVICE=4,1,3964R,"\\.\COM21",011
```

Operation at port 21, BFA monitoring 2s, high priority, with BCC

```
DEVICE=4,2,3964R,"\\.\COM22",100
```

Operation at port 22, BFA monitoring 30s, low priority, without BCC

```
DEVICE=4,3,3964R,"\\.\COM23",110,500,200,100
```

Operation at port 23, BFA monitoring 30s, with high priority,
without BCC, max. response time 500ms, output delay 200ms,
input delay 100ms.

DUST3962
computer interfacing
Gateway function

DEVICE=*anr*,0,3962,"SOCKET","*host*",*port*,*timeout*

The gateway function to the DUST 3962 can be used to connect two emulator systems.

anr Connection point number of the device according to ORG generation (decimal number).

host Name of the connected peer system (for the active connection partner). Not specified for the passive connection partner.

port Port number/socket number in the local computer (e.g. 22220)

timeout Monitoring time

Example

```
DEVICE=3,0,3962,"SOCKET","system1",22220,0;
```

active side at connection buildup; enter host name

```
DEVICE=3,0,3962,"SOCKET","",22221,0;
```

passive side at connection buildup; empty string

DUST3964R
computer interfacing
via TCP/IP

DEVICE = *anr*,*gnr*,3964R,"SOCKET",port-own,
host-remote:port-remote

This parameter defines the emulation of a DUST3964R computer interface connection via TCP/IP and LAN or RAS. A LAN/RAS network adaptor and TCP/IP must be installed in Windows (see also section 6.9.3).

anr Connection point number of the device according to ORG generation (decimal number).

gnr Device number according to ORG generation (decimal number).

port-own Port number/socket number in the local computer (e.g., 22223).

host-remote Host name or address in the remote computer
(e.g., winnt7 or 220.250.1.155).

port-remote Port number/socket number in the remote computer (e.g., 22225).

Example

```
DEVICE=1,1,3964R,"SOCKET",22223,winnt7:22225;
DEVICE=1,1,3964R,"SOCKET",22223,220.250.1.155:22225;
```

DUST3965R
computer interfacing
via DF32/DF42
or TCP/IP

DEVICE = *anr,gnr,*3965R,"*driver*",*kenn1,kenn2,size,options*

This parameter defines the emulation of a DUST3965R computer interface connection via DF32/DF42 module interface. The protocol driver MSV2 must be installed for the DF32/DF42 module. In this case the DF32/DF42 processes the MSV2 protocol so that the description and parameter definition of this module applies.
Alternatively, the DUST3965R interfacing can be diverted to a TCP/IP interface connection via the Win socket interface using a standard LAN connection.

Important! Before the emulator is started, the following steps must be carried out with the COMSOFT program:

- Install the operating system on the module
- Install the protocol driver for the interface
- Define the protocol driver parameters

anr Connection point number of the device according to ORG generation (decimal number).

gnr Device number according to ORG generation (decimal number).

driver Select the device driver or operating mode:

"\\.\DFXX0"	for COMSOFT DF32/DF42
"SOCKET"	for TCP/IP transmission via Win socket

kenn1 for (*driver*) DFXX0: DF32/DF42 board number according to COMSOFT description.

for (*driver*) SOCKET: Host name or empty string

kenn2 for (*driver*) DFXX0: DF32/DF42 interface number according to COMSOFT description

for (*driver*) SOCKET: Port number (10000...)

size Transfer buffer size (as for COMSOFT driver installation)

options Option field (no bits used at present)

Examples

```
 device=1,1,3965R,"\\.\DFXX0",0,0,512;
device=1,1,3965R,"SOCKET","WINNT1",22220,512;
```

DUST3966
computer interfacing
via DF32/DF42
or TCP/IP

```
DEVICE = anr,gnr,3966,"driver",kenn1,kenn2,type,0,frame,t1,
         t2,t3,parfile
```

This parameter defines the emulation of a DUST 3966 computer interface connection via DF32/DF42 module interface. An HDLC UNC-2 or HDLC BAC 2.8 protocol driver must be installed for the DF32/DF42 module. In this case the DF32/DF42 processes the HDLC protocol so that the description and parameter definition of this module applies.
Alternatively, the DUST3966 interfacing can be diverted to a TCP/IP interface connection via the Win socket interface using a standard LAN connection.
In addition to this device entry in the *rpar.sys* configuration file, the DUST 3966 emulation requires a parameter file (*parfile*) that allows you to define additional operating modes. Please refer to the description in section 6.9.

anr,gnr as above

driver Select the device driver or operating mode:

"\\.\DFXX0"	for COMSOFT DFxx
"SOCKET"	for TCP/IP transmission via Win socket

kenn1 for (*driver*) DFXX0: DF32/DF42 board number according to COMSOFT description
for (*driver*) SOCKET: Host name or empty string

kenn2 for (*driver*) DFXX0: DF32/42 interface number according to COMSOFT description
for (*driver*) SOCKET: Port number (10000...)

type

B:	balanced
UP:	unbalanced primary
US:	unbalanced secondary

frame

for (*driver*) DFXX0:	0
for (*driver*) SOCKET:	Max. FRAME length

t1

0:	No input monitoring
1:	Input monitoring 1s (message)
5:	Input monitoring 5s (message)
24:	Input monitoring 24s (message)
101:	Input monitoring 1s (message/terminate connection)
105:	Input monitoring 5s (message/terminate connection)
124:	Input monitoring 24s (message/terminate connection)

t2 0: Don't terminate dial connection during transmission pauses
1: Terminate dial connection during transmission pauses

t3 Acknowledgement monitoring time in milliseconds

parfile For DFXX0, path and name of the parameter file acc. to 5.3.

Notes

- At present, 0 must be set for timers t1, t2 and t3!
- Although the *type* entry is of no relevance for socket operation, it must be specified.
- TCP/IP for point-to-point connection only.

Example

```
device=1,1,3966,"\\.\DFXX0",0,0,US,0,0,0,0,
g:\sicomp\3966us.par;

device=1,1,3966,"SOCKET","WINNT7",22227,US,1024,0,0,0;
```

DUST 3961
computer interfacing
Gateway function

DEVICE=*anr,gnr,*3961,"SOCKET",*ownhost,[shost:]port1[/port2], [rhost:]port1[/port2],frame,timeout;*

The gateway function to the DUST 3961 can be used to connect two emulator systems.

anr,gnr	as above
ownhost	Local IP address or host name (surrogate name) as specified in *...\system32\...\hosts* for the identification of the local network card. If the computer has only one NetCard, a "**.**" character can be entered here. See section 6.8 for a detailed description of all parameters.
shost	Host name for TCP/IP connection 'Transmit'
port1 *port2*	Port number 1 for TCP/IP connection 'Transmit' Port number 2 for TCP/IP connection 'Transmit'
rhost	Host name for TCP/IP connection 'Receive'
port1 *port2*	Port number 1 for TCP/IP connection 'Receive' Port number 2 for TCP/IP connection 'Receive'
frame	Max. data(transfer) size in bytes
timeout	Connection monitoring time in seconds

Example

```
device=8,0,3961,"SOCKET",PC-SICOMP,SICDEMO1:22227,
          SICDEMO2:22229,1024,10;
```

5.1.5 Additional Devices

Promea **timer**

DEVICE = *anr,gnr*,ZIG1[,"*timeopt*"]
DEVICE = *anr,gnr*,ZIG1X[,"*timeopt*"]
This parameter is used to define a Promea timer .

anr Connection point number of the device according to ORG generation (decimal number).

gnr Device number according to ORG generation (decimal number).

ZIG1 Standard setting (first *Set date/time* call is discarded).
ZIG1X First *Set date/time* call is executed,

timeopt If NT_TIME is specified, the Windows date/time is set every time the ORG date/time is set.
If SYS_TIME is specified, ORG interprets the date/time as absolute UTC time (coordinated world time, formerly referred to as GMT), independent of the time zone and summer/winter time.

Note

When ORG reads the date/time, the emulator reads the Windows date/time. If the date/time is changed via ORG, the emulator changes the Windows date/time only if the pathname NT_TIME is specified. If ORG date/time and Windows date/time shall run synchronously, pathname NT_TIME must be specified (e.g., for the operation of a radio clock). A time of day adjustment (e.g., from summer to winter time) must always be initiated by ORG (serial radio clock to PROMEA or manual time adjustment).

□ Automatic summer/winter time change must always be deselected.

Example

```
DEVICE=1,1,ZIG1;
DEVICE=1,1,ZIG1,"$NT_TIME$";
DEVICE=1,1,ZIG1X,"$SYS_TIME$";
```

3691A timer

DEVICE = *anr,gnr,*ALE1

This device defines a 3691A timer.

anr Connection point number of the device according to ORG generation (decimal number).

gnr Device number according to ORG generation (decimal number).

Example

```
DEVICE=1,3,ALE1;
```

Note

If the 3691A timer is the only peripheral device in the SICOMP-R system, the device name ALE1 must be changed to ALE1ONLY.

Example

```
DEVICE=1,3,ALE1ONLY;
```

Note

If the 3691-A is the only timer in the emulated ORG and the timer has no battery backup, it can be converted to a battery-backed timer without changes to the emulated ORG.
For DEVICE=... , enter ALE1ONLYSP instead of ALE1ONLY.

Example

```
DEVICE=1,3,ALE1ONLYSP;
```

Pseudo device for **PSD**

`DEVICE` = *`anr,gnr,`*`FTNT,"`*`pipename",length[,buffersize,orgerror]`*

This parameter defines a pseudo device for PDS communication (pipe to SICOMP data). In the SICOMP M system, a visual display unit DSSE/DSSA must be generated to implement this function. M2000 (*sir_ftnt.exe*) creates a *named pipe* in Windows for each processing direction and takes over the function of a pipe **server**. The data reading pipe receives the ending **`in`**. The data writing pipe receives the ending **`out`**.

`anr` Connection point number of the device according to ORG generation (decimal number).

`gnr` Device number according to ORG generation (decimal number).

`pipename` "tnt0" to "ftnt9“ are the available pipe names for the use with *m2psapi.dll* (see also PDS documentation). M2000 creates two pipes for every pseudo device - one with the ending **`in`** and one with the ending **`out`** - so that a pipe for each direction is available. If you work without PSD calls, pipe number assignment is not restricted.

`length` Pipe length in kilobytes. The entry 0 is replaced by the default value of 4 kB.

`buffersize` Size of the input buffer in the user program. The input buffer made available by the user program is filled up to the *buffersize* length and terminated with ETX.

`orgerror` Specified and = 0: output calls of the user program are always terminated without indication, even in the event of transmission errors.
Specified and '1': standard indication.

`DEVICE`=*`anr,gnr,`*`FTNTX,"`*`pipename",length[,buffersize,orgerror]`*

This parameter defines a pseudo device for PSD communication (pipe to SICOMP data). In the SICOMP M system, a visual display unit DSSE/DSSA must be generated to implement this function. M2000 (*sir_ftnt.exe*) makes use of an existing *named pipe* in Windows and takes over the function of a pipe **client**. The data reading pipe receives the ending **`out`**. The data writing pipe receives the ending **`in`**.

`anr,gnr` as above

`pipename` as above; the remote computer is specified, a pipe is not created.

`length` as above
`buffersize` as above
`orgerror` as above

Important! Standard timeout for "Call Input monitoring"
(*default value: 10s*).

Note

'Remote - Pipes' allows two SICOMP systems to exchange data directly with each other in M2000. FTNT is set up on computer 1 and FTNTX on computer 2. This way, data can be read out with ‘’STAUAL pseudo device’’ on computer 1, and read in with ‘’STEIAL pseudo device’’ on computer 2 (and vise versa). The Windows function Remote - Pipe handles the transmission over the computer network.
In ORG-PV, the DSS connections used as pseudo devices should be generated with <u>automatic minus acknowledgement</u>. Changes in the <u>generated</u> system can be carried out with TESTS - however, this should only be attempted by the experienced user.

Examples

```
DEVICE=2,0,FTNT,"ftnt0",0,1024,0   ;DSSE8 pseudo device for PSD;
DEVICE=2,3,FTNTX,"\\server1\pipe\ftnt0",8,1024,0
                                         ;DSSE3 pseudo device for PSD;
```

Pseudo device as
pipe server/client

`DEVICE``=`*`anr,gnr`*`,FTNTD,"`*`pipename`*`",`*`length[,buffersize,orgerror]`*
`DEVICE``=`*`anr,gnr`*`,FTNTDX,"`*`pipename`*`",`*`length[,buffersize,orgerror]`*

FTNTD creates a pair of pipes and processes the associated pseudo device as a **pipe server**.
FTNTDX establishes connections to a pair of pipes and processes the associated pseudo device as a **pipe client.**

A length word preceding the actual data is <u>not expected</u> by either of the two devices. The devices add a preceding length word to the user data when transmitting and remove it when receiving (see notes). All parameters similar as above.

anr,gnr	as above
pipename	as above
length	as above
Buffersize	as above
orgerror	as above

Important!

Standard timeout for "Call Input monitoring"
(*default value: 10s*)

Note

To enable the SICOMP software to communicate with pseudo devices, PSD provides the two pseudo device parameters FTNTD and FTNTDX. The convention ‘’length word" (see below) can be ignored for these parameters (see parameter description above).
Users <u>not</u> working with the standard program *PSDXXX* must observe the following programming convention in Windows, ORG-M, ORG-PV or ORG-HV: a word (SICOMP format) must precede the actual data identifying the data's overall length in words (length word). This convention results from the telegram interface between *PSDXXX* and PSAPI.

Examples

```
DEVICE=2,0,FTNTD,"ftntd0",0,512,0 ;DSSK8 pseudo device for PSD
DEVICE=2,3,FTNTDX,"\\server1\pipe\ftntd0",8,512,0
                                         ;DSSK3 pseudo device for PSD
```

Pseudo device for **WINCC-PSD**

`DEVICE` = *`anr,gnr`*,`WINCC`,"*`pipename", length[,buffersize,orgerror]`*

This parameter defines a pseudo device for WINCC-PSD. In the SICOMP-R system, a visual display unit DSSE/DSSA must be generated to implement this function. M2000 creates a *named pipe* in Windows for every processing direction and takes over the function of a pipe **server**. The data reading pipe receives the ending **`in`**. The data writing pipe receives the ending **`out`**.

`anr` Connection point number of the device according to ORG generation (decimal number).

`gnr` Device number according to ORG generation (decimal number).

`pipename` Any pipe name can be specified. M2000 creates two pipes for every pseudo device - one with the ending **`in`** and one with the ending **`out`** - so that a pipe for each direction is available.

`length` Pipe length in kilobytes. The entry 0 is replaced by the default value of 4 kB.

`buffersize` Size of the input buffer in the user program. The input buffer made available by the user program is filled up to the *buffersize* length and terminated with ETX.

`orgerror` Specified and = 0: output calls of the user program are always terminated without indication, even in the event of transmission errors.
Specified and '1': standard indication.

.

Example `DEVICE=2,0,WINCC,"wincc0",8,1024,0` ; pseudo device for PSD

Note As soon as *PSDXXX* or *WCCXXX* receive a write instruction for the symbolic CB name "@@@@@@" or the object number "9999", the user subroutine *UPAWP* is called with the associated parameters "address" and "value", which must be linked to *PSDXXX* or *WCCXXX*. Write instructions must always be transmitted for individual words. Read instructions are not permissible.

For users of WinCC and WinCC-PSD channel DLL, this means that an external PSD variable of type 'A' (unsigned 16-bit value) can be defined and assigned to the CB name "@@@@@@" and the address "0"(=V128). If the value "100" is written to this variable (WinCC function "SetTagWord"), the *UPAWP* subroutine is called in ORG-M or ORG-PV with G6:=0 and G7:=100. This allows users to activate any (self-programmed) function from within WinCC under the emulated ORG. The criteria used are the V address in "pseudo CB" 9999 and the transmitted value.
PSD variables assigned to CB "@@@@@@" may not be updated by WinCC!
The standard elements *PSDXXX* and *WCCXXX* contain a dummy module. This dummy module with the designation *PLSKx-PSD.PSDAWP* is supplied on the virtual disk PLSK.PSD and can be used as a template (it simply indicates the values transmitted for G6 and G7 on the standard signaling device).

Example

ORG shall be booted with and without start list via WinCC buttons. CB "@@@@@@" and address "3" (=V130) are assigned to the PDS variable "ORGBOOT" (unsigned 16-bit value). One button sets the value 1 (SetTagWord (ORGBOOT,1)), another button the value 2 (SetTagWord (ORGBOOT,2)). *UPAWP* first checks whether the "Boot ORG" function is meant by comparing G6 with 3. It then verifies via G7 (1 or 2), whether ORG is booted with or without start list and finally executes the function accordingly ($RUFORG"...).

Note

When old linker cards are used with new basic language elements, error message "/NDEF:UPAWP" is displayed upon loading *PSDXXX* and *WCCXXX*. This error message can be ignored as long as the new function is not used.

Dummy module

```
/#PSDAWPXX                         170800/PJG
************************************************************************
***        PSDXXX:   USER FUNCTION
************************************************************************
           'PKEN' ,,,,0
           'DEEX' UPAWP
           'NDEX' STAMELD

           'NAME' PSDAWP
           'RECORD' PSDAWP

'VA' TEXTA/
' Y'       = 'Z=PSDAWP: DUMMY: #### ####(3)'
'VA' TEXTE/

'IA' UPAWP/   ('R2) := R7          *** DUMMY!
' A'       ('R2) := R3       ***
' A'       ('R2) := R4       ***
' A'       ('R2) := R5       ***

' A'       *** G6            *** RELATIVE ADDRESS (0 and above)
' A'       *** G7            *** CURRENT VALUE

' A'       R3 := <7+TEXTA>           ***
' A'       R4 := 'H=8000'            ***
' A'       R5 := G6                  ***
' A'       $BIHEX;                   ***              N
' A'       'IA'                                       B
           ='H=E000'                                  B
            ***                                       B
' A'       R3 := <10+TEXTA>          ***
' A'       R4 := 'H=0000'            ***
' A'       R5 := G7                  ***
' A'       $BIHEX;                   ***              N
' A'       'IA'                                       B
           ='H=E000'                                  B
            ***                                       B
' A'       R5 := (R2')        ***
' A'       R4 := (R2')        ***
' A'       R3 := (R2')        ***

' A'       :SP (R2')          ***

'END'
```

SpecialPromea
MX1

`DEVICE` = *`anr,gnr,`*`EAMX1,"`*`pathname`*`",`*`time,bytes,flags`*

EAMX1 solely processes inputs transmitted to ORG via the serial interface. Input end criterion is either the elapsed monitoring time after the last character has been received, or the reached maximum limit. It is possible to define an ETX complement for the received data.

`anr` Connection point number of the device according to ORG generation (decimal number).

`gnr` Device number according to ORG generation (decimal number).

`pathname` Designation of the serial interface defined in Windows (COMx or \\.\COMxx).

`time` Monitoring time in milliseconds. The accumulated data is transmitted to ORG if no further characters arrive during this period after the last character has been received. A '0' entry deactivates time monitoring function.

`bytes` Max. number of characters. The accumulated data is transmitted to ORG as soon as this maximum limit is reached. A '0' entry deactivates the count monitoring function.

`flags` Parameters
Bit 0 = 0 Transmit data to ORG in unchanged form.
Bit 0 = 1 Transmit data to ORG with ETX complement.
Bit 1 = 0 Transmit data to ORG as byte stream.
Bit 1 = 1 Transmit data to ORG as word stream.
Bit 3 = 0 Set serial port standard parameters.
Bit 3 = 1 Do not define serial port parameters - the MODE command applies .
Bit 4 = 0 Accept characters in unchanged form and transmit to ORG.
Bit 4 = 1 Only transmit the 7 low-order bits to ORG.

Example

```
DEVICE=3,0,EAMX1,"\\.\COM13",70,20,0x9;
```

Inputs via serial port 13; monitoring time 70 ms;
buffer transmission to ORG upon reaching 20 bytes maximum, data with ETX complement, serial port is set via the MODE command.
(0x9 = bit 0 and bit 3 are set)

BDE terminal
ES

DEVICE = *anr,gnr*,EAES,"*pathname*",*options*

This parameter defines the connection of a BDE terminal of the ES 1 device family via serial interface.

Note The serial port must be set with the MODE command (*setall.bat*) before the emulator is started.

anr Connection point number of the device according to ORG generation (decimal number).
gnr Device number according to ORG generation (decimal number).

EAES Device designation

pathname Designation of the serial interface defined in Windows (COMx or \\.\COMxx).

options Parameters
Bit 4 = 0 Input not transparent
Bit 4 = 1 Input transparent
Bit 5 = 0 Output not transparent
Bit 5 = 1 Output transparent
Bit 6 = 0 Operation via run procedure
Bit 6 = 1 Operation via protected 38xx procedure
Bit 7 = 0 Must be '0', otherwise IPKS test mode is active

Examples

```
DEVICE = 1,1,EAES,"\\.\COM2",0;
```

Input and output not transparent, operation via run procedure.

```
DEVICE = 1,1,EAES,"\\.\COM2",0x40;
```

Input and output not transparent, operation via protected 38xx procedure.

```
DEVICE = 1,1,EAES,"\\.\COM2",0x70;
```

Input and output transparent, operation via protected 38xx procedure.

Parameter **EAU** EAU

If EAU is specified, commands to EA-U modules are not just passed through but are transferred to the associated device process.

Emulation
KS N16

DEVICE = *anr*,0,CS275,"\\.\NATDEV",*zyklus*

This parameter defines the connection to TELEPERM M systems
(connection to a CS275 bus via the KS N16 module).

Note The KS N16 module always provides two interfaces to ORG - this is why value '0' is specified instead of the device number 'gnr'. SIR_CS27.EXE always processes device '0' of the defined connection point number as output device, and device '1' of the defined connection point number as input device.

anr Connection point number under which KS N16 is generated in the original system (decimal number).
zyklus Read cycle in milliseconds
(entry '0' sets the default cycle of 10ms).

Example

```
DEVICE = 7,0,CS275,"\\.\NATDEV",300;
```

6 Device Emulation Details

6.1 DS074 Device Emulation

6.1.1 Screen Structure

The DS074 data station is emulated as a console application in a Windows window.

The following conditions apply:

- The name defined for *pipename* is displayed as the window header.
- A 9x15 character set is used in Window representation by default (Standard Window Screen).
- Full-screen mode can be set by clicking the right mouse button and selecting Screen Parameters–*Screen Size* (Full Screen).
- The screen area comprises:
  the data section (lines 1 to 24) and the status bar (line 25).
  The status line displays the tab stops and the current operating states (roll mode, block/character mode, virtual console).

The following operating modes are implemented at present:

Display in status bar	Roll	Roll mode Page mode	Shift+F6
	Block Zeich	Block mode Character mode	Shift+F7
SIC-M only	VK	Virtual console ON Normal operation	Shift+F8

You can use the key combinations Shift+F6, Shift+F7 and Shift+F8 to switch between the individual operating modes.

Scroll up/down screen pages (TE_ALPHA only)	Shift + UP ARROW key Shift + DOWN ARROW key	move to next screen page move to previous screen page

The number (number 2-6) of the current screen page is displayed in the status bar of the terminal emulation.
A detailed description of the individual functions is given in the User Manual Terminal Emulation.

6.1.2 Key Assignment

Enter	DUEZ
End	DUE
Ctrl End	DUEM
Ctrl B	STX
Ctrl C	ETX
Ctrl G	BEL
Ctrl L	FF (delete screen, move cursor to beginning of screen)
Ctrl U	Reset visual display unit (block and mask mode)
Backspace	Cursor left
Tab	Tab right
Shift Tab	Tab left
Ctrl Tab	Set tab stop
Ctrl Shift Tab	Reset tab stop
Ins	Insert character
Del	Delete character
+	Insert line (plus key on numeric pad)
-	Delete line (minus key on numeric pad)
Esc	KT '0'
F1	KT '1'
to	
F9	KT '9'
F10	KT ' : '
F11	KT ' ; '
F12	KT '<'

Alt 0	KT '0'
to	
Alt 9	KT '9'
Alt A	KT 'a'
to	
Alt Z	KT 'z'
Alt .	KT ' . '
Alt Shift A	KT 'A'
to	
Alt Shift Z	KT 'Z'
	Change operating mode:
Shift F6	Roll mode ON/OFF
Shift F7	Switch between block and character mode
Shift F8	Virtual console ON/OFF
	Scroll screen pages:
Shift ↑	Next screen page
Shift ↓	Previous screen page
	Character mode only:
End	Data transfer (DUE)
Ctrl Enter	Saving without termination (DUEZ)
Enter	LF (move cursor to beginning of next line)
Ctrl End	Termination and saving (DUEM)
PgUp	Scroll up (Ctrl R)
PgDn	Scroll down (Ctrl N)

Note The key assignment is defined through the parameter file *dsskx.tko* file. The key assignment shown above corresponds to the parameter file delivered with M2000.

6.2 DU05 Computer Interfacing

The DU05 device emulation offers two operating modes:

- DU05 emulation via DF42 transmission module
- Diversion of the DU05 communication to TCP/IP

6.2.1 DU05 Emulation via DF42

The DU05 device emulation processes the MSV2 transmission protocol. The following interface connections can be implemented:

- M2000 - M2000 via MSV2 and SINEC
- M2000 - External systems via MSV2

The WIKO functions are not implemented (Remote loading...).

6.2.1.1 Requirements

The software for the operation of the used hardware components is implicitly installed by M2000.

In addition to the device entry (*device* = ..) in the configuration file *mpar.sys*, DU05 device emulation parameters must be specified in a parameter file whose name and directory can be freely selected and is allocated in the device entry.

6.2.1.2 Parameter Definition

Note! All interface connections operated via COMSOFT modules must be parameterized with the COMSOFT program 'DFTEST' prior to the start of the emulation.

The application-specific parameters are stored in an ASCII file. It is possible to use both lower case and upper case letters in the parameter lines. A comment is introduced using the '#' character and extends to the end of the line.

The following parameters can be defined. The parameter details (including the default settings) are taken from the description of the associated COMSOFT driver:

PATH *pathname*	Directory with COMSOFT components
BOARD *no*	Board number
UNIT *no*	Interface number on the board
DU_MODUS *value*	0x00
OSFILE	OS42.BIN

6.2.1.3 MSV2 Protocol Parameters

The following protocol parameters can be defined for the MSV2 driver:

BAUDRATE	value
MODEM_SET	value (0..1)
MODEM_TO	value (1..255)
BITS	0x00
T1	value
T2	value
T3	value
T4	value
MZ1	value
MZ2	value
MZ3	value
MZ4	value
MULTI	value

6.2.2 Diverting DU05 Communication to TCP/IP

In the *mpar.sys* file of one of the two participating computers, the host name of the peer computer is entered as defined in the file*...\system32\drivers\etc\hosts*. This computer will be the active computer at connection buildup. In the *mpar.sys* of the other computer, the host name is an empty string. This computer will be the passive computer at connection buildup.
The data is transmitted on connection-oriented sessions based on the TCP Level 4 transport protocol.
The port number must be greater than/equal to 10 000 and is identical on both computers.

Specific startup functions are not required.

6.3 DU06 Computer Interfacing

The DU06 device emulation offers two operating modes:

- DU06 emulation via DF32 or DF42 transmission module
- Diversion of the DU06 communication to TCP/IP

6.3.1 DU06 Emulation via DF32/42

The DU06 device emulation processes the transmission protocols HDLC U/P, HDLC U/S and HDLC B.

The following interface connections can be implemented:

- M2000 - M2000 via HDLC B or HDLC U and SINEC
- M2000 - IBM via SDLC and SINEC SNSNA
- M2000 - X.25 network via HDLC B and SINEC SNPV.

The WIKO functions are not implemented (Remote loading...).

6.3.1.1 Requirements

The software for the operation of the used hardware components is implicitly installed by M2000.

In addition to the device entry (*device* = ..) in the configuration file *mpar.sys* (see 4.2.4 Interface Connections), DU06 device emulation parameters must be specified in a parameter file whose name and directory can be freely selected and is allocated in the device entry.

The processing of the function '**SIM**' must be separately defined in the DFXX driver (the definition of these parameters is not supported by the emulator).
A DFXX module used as 'DU06 unbalanced' with **SIM** support by the emulator must be parameterized with the program DFTEST prior to the start of the emulation.
The DU06 parameter file (see next page) must contain the entries BOARD, UNIT, and PATH as well as the line 'DU_MODUS 0'. All other entries will then be ignored.

6.3.1.2 Parameter Definition

The application-specific parameters are stored in an ASCII file. It is possible to use both lower case and upper case letters in the parameter lines. A comment is introduced using the '#' character and extends to the end of the line.

The following parameters can be defined for both protocol types. The parameter details (including the default settings) are taken from the description of the associated COMSOFT driver (HDLC-UNC or HDLC-BAC):

PATH *pathname* Directory with COMSOFT components
BOARD *no* Board number
UNIT *no* Interface number on the board
DU_MODUS *value* Bit-by-bit adjustment of special functions by means of a 32-bit value:

- Bit 0 = 0 Drivers and parameters are assigned once to the module when the emulation is started.
- Bit 0 = 1 Drivers and parameters are assigned to the module every time ORG initiates basic parameter definition.
- Bit 1 = 0 No 3277 connection to IBM
- Bit 1 = 1 3277 connection to IBM
- Bit 2 = 0 An HDLC connection is terminated exclusively with the COMSOFT driver instruction 'CLOSE_UNIT'.
- Bit 2 = 1 An HDLC connection is terminated with the COMSOFT driver instruction 'CLOSE_UNIT' and subsequent driver parameterization.

```
# If a DF42 module is used, the following line must be activated:
# OSFILE OS42.BIN
```

The operation of a DU06B, DU06UP or DU06US via a COMSOFT module requires (at present) that bit 2 is set in the parameter DU_MODUS to ensure that the physical connection is terminated after the buildup of the logic connection (adaptation to the current behavior of the COMSOFT drivers).

6.3.1.3 HDLC-UNC Protocol Parameters

With COMSOFT drivers DHDLC.DRV C01 and higher, it is possible to process modem signals when operating a DU06UP or DU06US via a COMSOFT module. To implement modem operation, the new parameters MODEM and MODEM_TIMEOUT must be defined. In addition to the COMSOFT driver DHDLC.DRV, the COMSOFT components DHDLC.DAT, OS32.BIN and DFXX0 too must be integrated in the Windows system.
DFXX0 is installed in the *%systemroot%\system32\drivers* directory (Windows must be restarted).

The following protocol parameters can be defined for the HDLC-UNC driver:

MODEM	YES│NO	
MODEM_TIMEOUT		
SNRMREP	*value*	(0..255)
NRZ	YES│NO	
CRC	0│1	
MASTER	YES│NO	
CLOCK	*value*	(0..3)
MULTIPOINTYES│NO		
CREDIT	*value*	(0..7)
IFRAMESIZE	*value*	(128..1024)
STATIONS	*value*	(1..32)
HDLCADR	*adr*	
TIMER1	*value*	
TIMER2	*value*	
BAUDRATE	*value*	

6.3.1.4 HDLC-BAC Protocol Parameters

The following protocol parameters can be defined for the HDLC-BAC driver:

DCE	YES│NO	
MODEMDEF	*value*	(0..15)
MODEM_I_TO	*value*	(0..255)
MODEM_I_OK	*value*	(0..255)
AUTORESTART	YES│NO	
SENDSABM	YES│NO	
REP	*value*	(0..255)
NRZ	YES│NO	
CRC	0│1	
RNRBRK	*value*	(0..255)
CLOCK	*value*	(0 │ 1 │ 4)
CREDIT_HDLC	*value*	(0..7)
V24V11	0│1	
IFRAMESIZE	*value*	(64..1024)
HDLCADRCON	*adr*	
HDLCADROWN	*adr*	
TIMER1	*value*	
TIMER2	*value*	
BAUDRATE	*value*	

6.4 Open DU0x Communication Interface

6.4.1 Overview

The open DU0x communication interface of the M2000 emulator (type SICOMP-M) allows you to divert the inputs/outputs transmitted via a DU0x module generated in an emulated ORG-M system to a TCP/IP socket connection. This way, ORG-M applications can exchange data with all systems that communicate via TCP/IP and provide one or several peer programs based on the same specifications.
It is also possible to interlink two emulated ORG-M systems via this interface. One system will always be the active system at connection buildup while the peer system remains passive. Once the connection is established, both systems can transmit and receive freely.

Note The representation of numerical values can differ for different hardware platforms, which must be taken into account when defining the user data to be exchanged. The described interface transmits all data unchanged in both directions. Numerical data from an application running under an emulated ORG-M system is always transmitted in SICOMP-M representation. If required, these data must be converted by one of the two participating applications.

6.4.2 Diverting DU04 Communication to TCP/IP

In the parameter files of the two participating computers, the host name or address of the peer system is entered as specified in the file*...\system32\drivers\etc\hosts*.
The data is transmitted in connectionless mode based on the UDP Level 4 transport protocol. Acknowledgement traffic should be implemented on the application level (as implemented in SINEC VS, for example).
This transmission can also be used to implement simple interface connections to external systems such as UNIX.
The port addresses must be unambiguous for each computer (a number greater than/equal to 10 000 must be specified).
Specific startup functions are not required.

6.4.3 Diverting DU05 Communication to TCP/IP

Connection buildup

Each of the two communication peers creates a TCP/IP socket (address family AF_INET, type specification SOCK_STREAM) through which all data and acknowledgements are exchanged.
On the emulator side, the socket options 'REUSEADDR' and 'KEEPALIVE' are selected.
The emulator can be defined both as active or passive system at connection buildup. The passive peer waits for the connection to be established (ACCEPT), while the active peer tries to establish the connection (CONNECT).
The user program on the peer system should work with the corresponding parameters (similar options selected, immediate connection buildup after program start or disconnection, etc.).

Connection cleardown/ disconnection

The emulator terminates the socket connection before the program ends or after the connection monitoring period has elapsed.

Data traffic

Transmission from the ORG-M application to the peer application

The ORG-M application initiates one outputs to the corresponding DU05 (call KOPAUSH). Only the call options 'Data', 'Code-oriented / transparent' and 'Block end ETX / block end ETB' (if applicable) may be used, but not 'Header', 'Header and data, 'Status', etc.

The emulator adds this transmission length as a prefix in the form of a data word to the actual data to be transmitted (INTEL representation).

The peer system returns an acknowledgement message confirming that it has received the data.

Data
(byte-by-byte representation)

Length of the user data in bytes (low byte)
Length of the user data in bytes (high byte)
User data... (first byte)
...
User data (last byte)

Positive acknowledgement
to the emulator
(byte-by-byte representation)

0xFF
0xFF

Negative acknowledgement
to the emulator
(byte-by-byte representation)

0xFE
0xFF

Transmission from the peer application to the ORG-M application

The ORG-M application waits for an input via the corresponding DU05 (call KOPEINH).

Upon data transmission via TCP/IP, the DU05 device process evaluates the first two bytes (transmission length or acknowledgement identifier). The received data are terminated with an ETX data delimiter and passed on to the waiting ORG-M application **without** the 'message length' word. A positive acknowledgement is then returned to the peer system.

6.4.4 Diverting DU06 Communication to TCP/IP

In the parameter file of one of the two participating computers, the host name of the peer computer is entered as specified in the file *...\system32\drivers\etc\hosts*. This will be the active computer at connection buildup. In the parameter file of the other computer, the host name is an empty string. This computer will be the passive computer at connection buildup.
The data is transmitted on connection-oriented sessions based on the TCP Level 4 transport protocol.
The port number is identical on both computers and must be greater than/equal to 10 000.

Specific startup functions are not required.

6.5 CP1400 Communication Control

6.5.1 Installation

The installation procedure for an H1 interface connection described in the following is predetermined by the installation sequence for the original SICOMP-M system. It is assumed that this installation sequence is known to the user.

A CP1413 or CP1613 module and the corresponding SIMATIC NET software must be installed on the computer running the M2000 system. Also, the parameter definition program NML must be installed under Windows to be able to generate a H1 database.
It is assumed that the so-called AP system for the use of the CP1400 is installed in ORG-M of the original SICOMP system.

Step 1: The CP1400 parameter files are generated with the program NML running under Windows (described in the CP1400 documentation). The generated files have the designations *xyz.ADB* and *xyz.LDB*. File *xyz.ADB* is then copied to file *APDBA0,* file *xyz.LDB* is copied to file *APDBA1*.

Step 2: File *xyz.LDB* is accessed by SIMATIC NET under Windows. An error message is displayed that must be acknowledged and the file *xyz.TXT* is generated. In *xyz.TXT*, the lines marked as faulty at the file end must be deleted using a text editor (such as NOTEPAD).

Step 3: The newly generated file *xyz.TXT* is accessed by SIMATIC NET under Windows. A binary file *xyz1.LDB* is generated that can be used as a template and adapted as required. The *xyz.LDB* file is then used as database for the CP1413 or CP1613.

Step 4: The M2000 system is started for the first time. The ORG system running under M2000 should start without start list. It should also be possible to access the "disk" PLSK.PSD delivered with M2000 from the ORG system. The programs PCCOPY and CP1400 are loaded to the ORG system (PRP or HRP) from the GSB library on this PLSK.PSD disk. PCCOPY is used to copy the files *APDBA0* and *APDBA1* generated under Windows in step 1 to the ORG disk that contains the databases of the AP system. After that the newly loaded program CP1400 is started (once) in ORG. The CP1400 program connects the programs of the AP system with the copied databases *APDBA0* and *APDBA1*.
The CP1400 program must be executed each time new databases are copied. It can also be started with every SICOMP system start prior to starting the AP system components.
The condition :APADMI:CHANGE STARTID TO ACTUAL does not apply.

In the last step, M2000 is closed and Windows is started. After the start of M2000 and the SICOMP system running under M2000, the CP 1400 works with the new parameters.

6.6 CS275 Communication Control

6.6.1 Module Installation

To install the NAT module in the computer, follow the procedure described in the included documentation.
It is not necessary to assign an interrupt to the module. A note should be made of the I/O address specified on the module (see *NatDev.ini*).

6.6.2 Software Installation

The M2-CS optional package (KS N16/CS275 communication control, connection to TELEPERM) contains a detailed description of the installation procedure.

See also: section 4.2.4 Interface Connections/CS275
section 5.1.5 Additional Devices/KS N16

6.7 DUST3965R Computer Interface Connection

The DUST3965R device emulation offers two operating modes:

- DUST3965R emulation via DF42 transmission module
- Diversion of the DUST3965R communication to TCP/IP

6.7.1 DUST3965R Emulation via DF42

The DUST3965R device emulation processes the MSV2 transmission protocol. The following interface connections can be implemented:

- M2000 - M2000 via MSV2 and SINEC
- M2000 - External systems via MSV2

The WIKO functions are not implemented (Remote loading...).

6.7.1.1 Requirements

The software for the operation of the used hardware components is implicitly installed by M2000.

In addition to the device entry (*device = ..*) in the configuration file *rpar.sys*, DUST3965R device emulation parameters must be specified in a parameter file whose name and directory can be freely selected and is allocated in the device entry.

6.7.1.2 Parameter Definition

Note! All interface connections operated via COMSOFT modules must be parameterized with the COMSOFT program 'DFTEST' prior to the start of the emulation.

The application-specific parameters are stored in an ASCII file. It is possible to use both lower case and upper case letters in the parameter lines. A comment is introduced using the '#' character and extends to the end of the line.

The following parameters can be defined. The parameter details (including the default settings) are taken from the description of the associated COMSOFT driver:

PATH *pathname*	Directory with COMSOFT components
BOARD *no*	Board number
UNIT *no*	Interface number on the board
DU_MODUS *value*	0x00
OSFILE	OS42.BIN

6.7.1.3 MSV2 Protocol Parameters

The following protocol parameters can be defined for the MSV2 driver:

BAUDRATE	value
MODEM_SET	value (0..1)
MODEM_TO	value (1..255)
BITS	0x00
T1	value
T2	value
T3	value
T4	value
MZ1	value
MZ2	value
MZ3	value
MZ4	value
MULTI	value

6.8 DUST3961 Computer Interface Connection, Gateway Function

Two TCP/IP sockets (address family AF_INET, type specification SOCK_STREAM) are created for each 3961 parameter module. The first socket transmits data to the peer system, the second socket receives data from the peer system.
The emulator sets the socket options REUSEADDR and KEEPALIVE, the size of the data field is set according to the emulator parameters.
For each socket the emulator can be defined both as the active or passive system at connection buildup. Upon starting, the emulator then tries to establish a connection either actively or passively. Existing connections remain active until one of the peers clears the connection or a transmission error occurs.
After a connection has been cleared or interrupted, the emulator immediately tries to re-establish the connection.
The first four bytes of the transmitted data define the net length of the corresponding message (32-bit value in Intel representation).
The 3961 gateway function adds this 32-bit value as a prefix to the data transmitted from the ORG system to the 3961 device.
When the peer system receives the data, the 3961 gateway function removes the 32-bit value before the data is passed on to the ORG system.

```
Device=anr,gnr,3961,"SOCKET",ownhost,transmit connection,
receive connection, data length, timeout
```

ownhost

Local TCP/IP address or TCP/IP surrogate name (alias) as specified in *...\system32\...\hosts* for the identification of the local network card.
If the computer has only one network card, a "**.**" character can be entered here.

transmit connection

Syntax: [*hostname:*]*port1*[/*port2*]
The defined connection transmits data to the peer system.
If the *hostname:* is specified, the connection to the peer system is set up actively. The local socket *ownhost:port1* is created and the connection to the port *hostname*:*port2* is established. If *port2* is not specified, *port2* is set to *port1*.
If the *hostname*: is not specified, the connection to the peer system is set up passively. The local socket *ownhost*:*port1* is created and the system waits for the connection to be established by the peer system. *port2* is not used for passive connections.

receive connection

Syntax: [*hostname*:]*port1*[/*port2*]
The defined connection receives data from the peer system.
If the *hostname*: is specified, the connection to the peer system is set up actively. The local socket *ownhost*:*port1* is created and the connection to the port *hostname*:*port2* is established. If *port2* is not specified, *port2* is set to *port1*.
If the *hostname*: is not specified, the connection to the peer system is set up passively. The local socket *ownhost*:*port1* is created and the system waits for the connection to be established by the peer system. *port2* is not used for passive connections.

data length

Number of bytes of the maximum amount of data transmitted

timeout

Connection monitoring time in seconds. If no data is transmitted during this period of *n* seconds, the transmit connection is terminated (and subsequently the receive connection is terminated by the peer system).

6.9 DUST 3966 Computer Interface Connection

The DUST3966 device emulation offers two operating modes:

- DUST 3966 emulation via DF32 or DF42 transmission module
- Diversion of the DUST 3966 communication to TCP/IP

6.9.1 DUST 3966 Emulation via DF32/42

The DUST 3966 device emulation processes the transmission protocols HDLC U/P, HDLC U/S and HDLC B.
The following interface connections can be implemented:

- M2000 - M2000 via HDLC B or HDLC U and SINEC
- M2000 - IBM via SDLC and SINEC SNSNA
- M2000 - X.25 network via HDLC B and SINEC SNPV.

The WIKO functions are not implemented (Remote loading...).

6.9.1.1 Requirements

The software for the operation of the used hardware components is implicitly installed by M2000.
In addition to the device entry (*device* = ..) in the configuration file *rpar.sys* (see 5.1.4 Interface connections), DUST3966 device emulation parameters must be specified in a parameter file whose name and directory can be freely selected and is allocated in the device entry.

6.9.1.2 Parameter Definition

The application-specific parameters are stored in an ASCII file. It is possible to use both lower case and upper case letters in the parameter lines. A comment is introduced using the '#' character and extends to the end of the line.

The following parameters can be defined for both protocol types. The parameter details (including the default settings) are taken from the description of the associated COMSOFT driver (HDLC-UNC or HDLC-BAC):

PATH *pathname*	Directory with COMSOFT components
BOARD *no*	Board number
UNIT *no*	Interface number on the board
DU_MODUS *value*	Bit-by-bit adjustment of special functions by means of a 32-bit value:
Bit 0 = 0	Drivers and parameters are assigned once to the module when the emulation is started.
Bit 0 = 1	Drivers and parameters are assigned to the module every time ORG initiates basic parameter definition.
Bit 1 = 0	No 3277 connection to IBM
Bit 1 = 1	3277 connection to IBM
Bit 2 = 0	An HDLC connection is terminated exclusively with the COMSOFT driver instruction 'CLOSE_UNIT'.
Bit 2 = 1	An HDLC connection is terminated with the COMSOFT driver instruction 'CLOSE_UNIT' and subsequent driver parameterization.

```
# If a DF42 module is used, the following line must be activated:
# OSFILE OS42.BIN
```

The operation of a DUST 3966B, DUST 3966UP or DUST 3966US via a COMSOFT module requires (at present) that bit 2 is set in the parameter DU_MODUS to ensure that the physical connection is terminated after the buildup of the logic connection (adaptation to the current behavior of the COMSOFT drivers).

6.9.1.3 HDLC-UNC Protocol Parameters

With COMSOFT drivers DHDLC.DRV C01 and higher, it is possible to process modem signals when operating a DUST 3966UP or DUST 3966US via a COMSOFT module. To implement modem operation, the new parameters MODEM and MODEM_TIMEOUT must be defined. In addition to the COMSOFT driver DHDLC.DRV, the COMSOFT components DHDLC.DAT, OS32.BIN and DFXX.SYS too must be integrated in the Windows system.
DFXX.SYS is installed in the *%systemroot%\system32\drivers* directory (Windows must be restarted).

The following protocol parameters can be defined for the HDLC-UNC driver:

MODEM	YES│NO	
MODEM_TIMEOUT		
SNRMREP	*value*	(0..255)
NRZ	YES│NO	
CRC	0│1	
MASTER	YES│NO	
CLOCK	*value*	(0..3)
MULTIPOINTYES│NO		
CREDIT	*value*	(0..7)
IFRAMESIZE	*value*	(128..1024)
STATIONS	*value*	(1..32)
HDLCADR	*adr*	
TIMER1	*value*	
TIMER2	*value*	
BAUDRATE	*value*	

6.9.1.4 HDLC-BAC Protocol Parameters

The following protocol parameters can be defined for the HDLC-BAC driver:

DCE	YES│NO	
MODEMDEF	*value*	(0..15)
MODEM_I_TO	*value*	(0..255)
MODEM_I_OK	*value*	(0..255)
AUTORESTART	YES│NO	
SENDSABM	YES│NO	
REP	*value*	(0..255)
NRZ	YES│NO	
CRC	0│1	
RNRBRK	*value*	(0..255)
CLOCK	*value*	(0 │ 1 │ 4)
CREDIT_HDLC	*value*	(0..7)
V24V11	0│1	
IFRAMESIZE	*value*	(64..1024)
HDLCADRCON	*adr*	
HDLCADROWN	*adr*	
TIMER1	*value*	
TIMER2	*value*	
BAUDRATE	*value*	

6.9.2 Diverting DUST 3966 Communication to TCP/IP

In the parameter file of one of the two participating computers, the host name of the peer computer is entered as defined in the file *...\hosts*. This computer will be the active computer at connection buildup. In the parameter file of the other computer, the host name is an empty string. This computer will be the passive computer at connection buildup.
The data is transmitted on connection-oriented sessions based on the TCP Level 4 transport protocol.
The port number must be greater than/equal to 10 000 and is identical on both computers.

Specific startup functions are not required.

6.9.3 Diverting DUST3964R Communication to TCP/IP

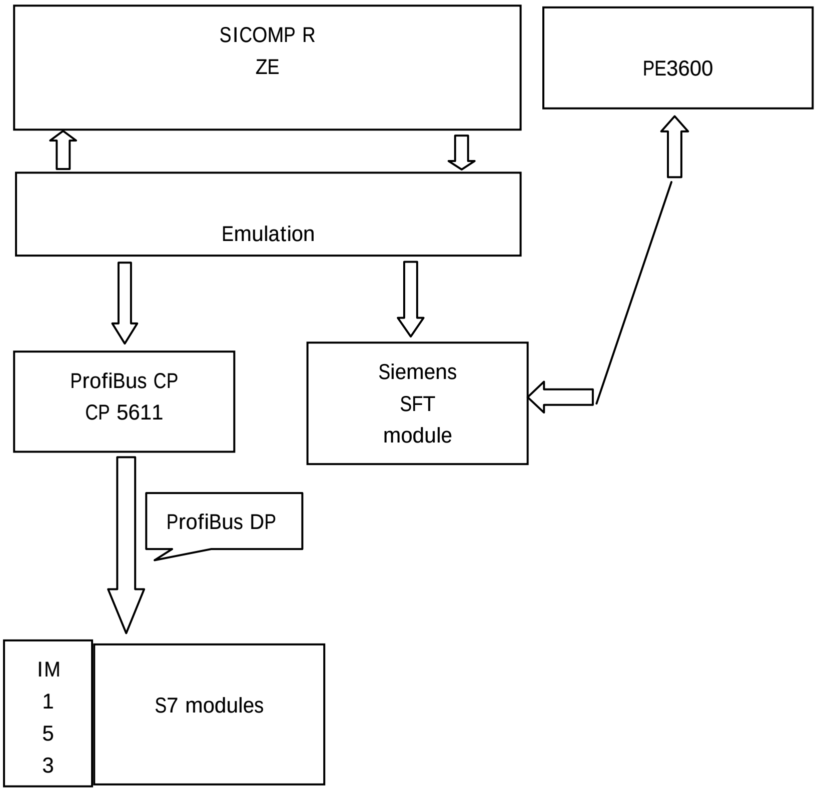
In the parameter files of the two participating computers, the host name or address of the peer system is entered as specified in the file *...\hosts*.
The data is transmitted in connectionless mode based on the UDP Level 4 transport protocol. Acknowledgement traffic should be implemented on the application level (as implemented in SINEC VS, for example).
This transmission can also be used to implement simple interface connections to external systems such as UNIX.
The port addresses must be unambiguous for each computer (a number greater than/equal to 10 000 must be specified).
Specific startup functions are not required.

6.10 PE3600 Emulation via ProfiBus or SFT Module

6.10.1 M2000

The SICOMP-R configuration is simulated in the M2000 system using the Siemens ProfiBus as interface to the process I/O equipment.
In the emulation, S7 modules implement the connection to the process I/O equipment - analogous to the PE3600 for the SICOMP-R system. A CP 5611 Profibus module is installed in the M2000 computer connecting Windows/M2000 to the S7 hardware.
Alternatively, a Siemens SFT module can be used to establish a direct connection between Windows/M2000 and the PE36000.

The resulting configuration is shown in the below diagram.



6.10.2 Configuration File *rpar.sys*

The process parameters are defined centrally by means of the M2000 configuration file *rpar.sys*. This file must contain the entries described below.

A ‘device’ entry to integrate PE3600 calls in the emulation.

`DEVICE` `=` *`anr,gnr,PE3600`*

`anr` Connection point number of the device (decimal number).

`gnr` Device number (decimal number).

An entry assigning the following process signal converters to a hardware group. This entry applies until another PSFDEV parameter is defined.

PSFDEV = *devtyp,devno,mode,polltime*

devtyp Module type
PROFI = Profibus module
SFT = SFT module

devno Module number
PROFI 1.... decimal
SFT 0...3 decimal

mode Mode, hexadecimal, '0' only is used at present

polltime Cycle time in milliseconds at which alarm generating (input) signal converters are scanned for changes.

An entry for every basic and expansion control unit as well as for every process signal converter and allocation of the module address (of the module defined earlier in the PSFDEV entry). The syntax is determined by the corresponding module and the signal converter type. The syntax below represents the presently implemented control units and signal converters allocated to a Profibus module.

Parameters for a GS3601-B basic control system:

PSF = GS3601-B,*sicAdr*,*AddressLevel,Spas,Reset,Poa*

Description of the parameters:

sicAdr Address of the control unit under SICOMP-R

AddressLevel Address level of the basic control unit. Possible entries are AE1 for address level 1 and AE2 for address level 2 (corresponds to switches S2 and S3).

Spas Addressing mode of the SP connection points. SPAS16 means that connection points 0 to 15 only can be addressed. SPAS22 means that connection points 16 to 21 too can be addressed. This entry corresponds to switch S1.

Reset This entry determines whether the control unit can be reset with ZK or not. Possible entries are ZKRESET and NORESET. This entry corresponds to switch S6.

Poa This entry determines whether the control unit can initiate a POA. or not. Possible entries are POA and NOPOA. This entry corresponds to switch S8.

Parameters for the process signal converters connected to the basic control unit:

Digital input DE3615-C

PSF = `DE3615-C,`*`sicAdr`*`,SlaveNo,OffSet,Alarm`

`sicAdr` Address of the process signal converter under SICOMP-R

`SlaveNo` Number of the Profibus slave.

`Offset` Offset address within the Profibus slave.

`Alarm` This entry determines whether the control unit can generate alarms and thus initiate a POA. Possible entries are ALARM and NOALARM. This entry corresponds to switch S8.

Digital output DA3621-B

PSF = `DE3621-B,`*`sicAdr`*`,SlaveNo,OffSet`

`sicAdr` Address of the process signal converter under SICOMP-R

`SlaveNo` Number of the Profibus slave.

`Offset` Offset address within the Profibus slave.

6.10.3 Example

The parameter file *rpar.sys* contains the following instructions:

```
device = 6,0,PE3600                             1.)

PSFDEV = PROFI,1,0,100                          2.)

PSF = 3601-B,0000,AE1,SPAS16,ZKRESET,POA       3.)

PSF = 3615-C,6000,3,0,ALARM                     4.)
```

1. This entry in the parameter file defines a PE3600 unit as device no. 0 at connection point 6.

2. This entry defines a Profibus module (number 1) with mode 0. This means that access of subsequently defined process signal converters (PSF entry) is conducted via this module. All alarm generating input modules are scanned at 100 millisecond intervals.

3. This entry defines a GS3601-B basic control unit under R-address '0' in address level 1. The SP address must be defined in the range from 0 to 15. Reset with ZK and peripheral control system requests are possible. All subsequently defined process signal converters are assumed connected to this control unit.

4. This entry defines a 32-bit digital input under R-address '6000H'. Access is mapped to Profibus slave 3 and the offset addresses 0 to 3. The module generates alarms (i.e. POA at changes).

6.11 PE3600 – Siemens SFT Module

6.11.1 Example

```
device = 6,0,PE3600                             1.)

PSFDEV = SFT,0,00000000                         2.)

PSF = 3601-B,0000,AE1,SPAS16,ZKRESET,POA       3.)
```

1. This entry in the parameter file defines a PE3600 unit at connection point 6, device number 0.

2. This entry defines an SFT module (number 0) with mode 0.

3. This entry defines a GS3601-B basic control unit under R-address '0' in address level 1. The SP address must be defined in the range from 0 to 15. Reset with ZK and peripheral control system requests are possible. Only one 'PSF = ...' instruction is required.

Note

Module C71458-6008-A... must be used in the basic control system 3601. A connection between SFT and PE3600 cannot be established otherwise.

6.12 Interface Concentrator

6.12.1 General

In SICOMP configurations, printers, screens and other peripheral devices are usually connected to the computer by means of serial lines, which in large plants results in a considerable amount of cabling.
With M2000, the amount of cabling can be reduced by locally combining several of these remote devices (e.g., on remote premises, machines, workstations). Proxy programs are available for the M2000 interface emulations DS075 (serial display unit connection), DR202 (serial printer connection) and EAMX (PROMEA MX serial connection) that can run on remote Windows systems communicating with M2000 via LAN. It is thus possible to relocate these serial interfaces to a remote Windows computer communicating with M2000 via LAN and connect the peripheral devices to these remote interfaces.

In the parameter file *mpar.sys* on the M2000 computer, the parameter line of every relocated interface must be changed accordingly; on the remote computer, a proxy process must be started for each one of the serial interfaces.

The assigned interface parameters are transmitted to the remote computer as soon as the connection is established and also after every reparameterization (e.g., after a restart of the SICOMP system). In disconnected mode, the local device processes behave like deactivated terminals.

For every relocated serial interface, **two** TCP/IP connections are set up between the M2000 computer and the remote computer (the M2000 computer is the passive system).

6.12.2 Parameters in *mpar.sys*

The device name for the devices DR202, DS075 and EAMX can also be "REM:y" instead of "\\.\COMxx". "y" represents the port number of the first TCP/IP socket created by the associated process on the M2000 computer. A second port with the port number y+1 is created.

Examples of *mpar.sys* entries:

```
DEVICE = C0C8, DS075, "REM:22000";
DEVICE = C020, DR202, "REM:23000";
DEVICE = C040, EAMX,  "REM:24000", 1, 0, 0;
```

6.12.3 Proxy Processes on the Remote Computer

Depending on the device type, one of the three new processes RC_DRUA.EXE, RC_DS075.EXE and RC_EAMX.EXE is started. The structure is identical for all three call instructions:

```
RC_DRUA.EXE \\.\COMxx SocketOwn Host:SocketCon
RC_DS075.EXE \\.\COMxx SocketOwn Host:SocketCon
RC_EAMX.EXE \\.\COMxx SocketOwn Host:SocketCon
```

SocketOWN Port number of the first local socket. Port SocketOwn+1 is also used.

Host Name of the M2000 computer acc. to file: *%systemroot%system32\drivers\etc\hosts*.

SocketCon Port number of the first socket used on the M2000 computer.

COMxx The serial interface on the remote computer.

In addition to the three processes above, the processes SIC_MELD.EXE and WAIT4.EXE should also be installed on the remote computer and included in the search path.

Process messages are displayed in a window and entered in the event log. A process is closed by closing the corresponding window or when the user logs off in WINDOWS.

Example of start batch on the remote computer:

```
REM
REM Wait for WINDOWS
REM
wait4

REM
REM Set up interface COM10 for DSSK20
REM
mode \\.\COM10 ...

REM
REM Start proxy process for DSSK20
REM
start “DSSK20“ /min rc_ds075.exe \\.\COM10 22000 PCSIC1:22000

REM
REM Set up interface COM11 for DRUA21
REM
mode \\.\COM11 ...

REM
REM Start proxy process for DRUA21
REM
start “DRUA21“ /min rc_drua.exe \\.\COM11 23000 PCSIC1:23000

REM
REM Set up interface COM12 for DRUA22
REM
mode \\.\COM12 ...

REM
REM Start proxy process for DRUA22
REM
start “DRUA22“ /min rc_eamx.exe \\.\COM12 24000 PCSIC1:24000
```

6.13 OP31 Emulation

6.13.1 Setup Instructions for SONY RMO-S580 Drive

6.13.1.1 Function Switch on the Rear Panel

A: Parity check
= 0: SCSI parity check
= 1: No SCSI parity check

B: Reserved

C: Write cache control
= 0: Activate write cache
= 1: Deactivate write cache

D: Manuel disk ejection
= 0: Ejection with EJECT button possible
= 1: Ejection with EJECT button not possible

E: SCSI1/SCSI2
= 0: SCSI2
= 1: SCSI1

F: Termination
= 0: Internal termination deactivated
= 1: Internal termination activated

G: Automatic motor start
= 0: Spindle motor starts when MO-DISK is inserted.
= 1: Spindle motor does not start when MO-DISK is inserted.

H: Reserved

6.13.1.2 Commissioning

- Connect power cable
- Connect SCSI cable
- Set SCSI-ID (rear panel)
- Terminate SCSI cable (if no other device is used, switch F=1)
- Turn on the drive
- Restart Windows

6.13.2 Important Notes

The MO drive must be turned on before Windows is started.

Windows assigns the next available drive letter to the MO device. If this drive has already been used for network drives, a connection to the network drive with this letter cannot be established. The connection must be canceled and then reestablished using another letter.

The device numbers are specified during ORG generation (without gap, starting with zero).

The system checks whether the specified physical drive is of the type *Removable Media*. If this is not the case, M2000 aborts the setup.

M2000 treats the MO drive as a physical drive. You must have administrator rights to access the drive.

Reassignment of administrator rights may become necessary after an extension of the M2000 installation by the OP31 option.

Important! Please note that the OP31 is generated as 'P20PSD' in M-ORG. Only if that is true, the volume can be ejected with DRIVE OFF. If the OP31 is generated as a hard drive, a volume switch is not allowed.

6.13.3 Data Media

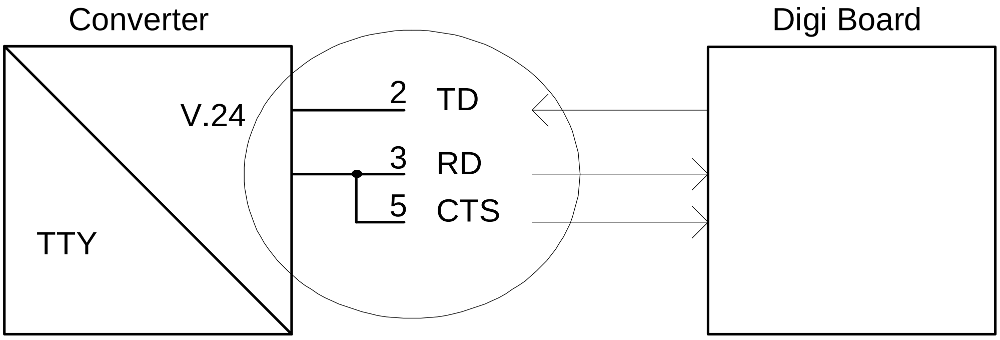
Only data media with 600MB, 512 byte/sector may be used. Appropriate Sony media have the designations:

- EDM-600B
- EDM-1DA1s

6.14 INOPERABLE Detection for Serial Interfaces

6.14.1 TTY / V.24 Interfaces

Operation via a TTY / V.24 converter requires the insertion of a jumper between pin 3 and pin 5 on the V.24 side (RD - CTS). The standard cables between the Digi Board and the converter are usually encapsulated so that the jumper must be inserted in the converter (an additional connection of the CTS signal from the converter must be avoided).



In the parameter file (*mpar.sys/rpar.sys*), INOPERABLE detection for MX devices, printers, DS075 and 3974R is set by means of the *flags* or *upar* parameter.

Attention! With the measures described above, INOPERABLE detection is ensured for the condition that the receive data line (seen from the Digi board side) changes to log. 0 (+12 Volt). In the event of a current loop interruption, this condition is fulfilled by the converter. However, if the cable between the Digi Board and the converter is removed (or if the converter fails), reliable INOPARABLE detection cannot be ensured as the RD input level remains undefined. Reliable detection can only be implemented by inserting the jumper in an adapter directly on the Digi Board (or in the Digi Board) and installing an additional pull-up resistor (RD against +12 Volt).

6.14.2 V.24 Connection

If the terminal equipment is directly connected to the V.24 interface, it must provide signal DSR (pin 6) available.
In the parameter file (*mpar.sys/rpar.sys*), the DSR scan for MX devices, printers, DS075 and 3974R is set by means of the *flags* or *upar* parameter.

Attention! With the measures described above, INOPERABLE detection is ensured for the condition that the DSR signal (seen from the Digi Board side) is not logic ON (+12 Volt). A removed slip-on lead will be detected. This solution cannot be implemented with standard printer cables (exception: DR 3918) as only TD, RD and ground are connected and thus an evaluable signal from the printer will not be available!

6.15 Compatibility Restrictions

6.15.1 SICOMP M Computer

M2000 replaces the SICMP M minicomputer in nearly all aspects. Although special emphasis has been placed on compatibility to the original system, some restrictions in regard to hardware compatibility must be accepted whose remedy would be too costly and in some cases even are desirable (see "time behavior" below).
These restrictions are described in the following.

Floating point values

During the transfer of floating point values from an emulated ORG system to the Windows system (usually via PSD/PSAPI), the mantissa are processed differently.
Original SICOMP K format: 56-bit mantissa
Microsoft DOUBLE format: 53-bit mantissa

Time behavior

Emulations based on today's modern PC hardware offer much more performance than the original SICOMP M with its out-dated components, with the result that SICOMP M application systems run much faster. Disk access in particular can be speeded up by factor 50 to 100 due to the high cache hit rates.
This "incompatibility" in the time behavior is a desirable side effect of an emulation (a well designed application software usually handles this increase in performance without problems).

Timer

Due to the use of standard hardware components and the Windows operating system, limitations with regard to time of day granuality and shortest possible cyclic wake-up time must be taken into account for time-modified calls. The shortest time unit is 10ms for ZIG1 and 1ms for ZIG2.

VICOM

The following VICOM commands are not implemented:

ACTIVATE	ACTIVATE CHECKPOINT
APZ	ACTIVATE PROCESS BY ‘APZ’
CALCULATE	CALCULATE TWO OPERANDS
DEACTIVATE	DEACTIVATE CHECKPOINT (S)
DELETE	DELETE CHECKPOINT (S)
DFP	DEACTIVATE PROCESS BY ‘DFP’
GENERATE	GENERATE ECC OF MEMORY
INFORM	ESTABLISH CHECKPOINT (NO STOP)
INTERRUPT	ESTABLISH CHECKPOINT (INTERRUPT)
PASSWORD	ESTABLISH OR DELETE PASSWORD
PROTECT	PROTECTION BY PASSWORD
SHOW	DISPLAY CHECKPOINTS
STEP	CONTINUE PROGRAM FOR STEP (S)
*	END OF CHANGE-INC-MODE

Test functions

TESTAS and SPAS are not supported. The command /STATI does not deliver any processable values as not all of the SICOMP ZE statistic registers can be mapped for performance reasons.

Floppy disk

The data structure of a floppy disks processed by the emulation corresponds to that of the original system. However, data exchange via this medium is not possible as today's floppy disk controllers no longer support the specific modulation of track 0.

DU06

The DU06 device emulation does not support the WIKO functions
(Remote loading ...).

Other restrictions

The following functions deviate:

- Volume switch (/DRIVEON and /DRIVEOFF)
- INITM (DELETE and DEL-PTR)

Volume switch (/DRIVEON and /DRIVEOFF)

A volume switch is not detectable under Windows.

To avoid loss of data, follow the below procedure:

- Lock the drive with the MO disk inserted (the MO disk cannot be changed).
- The lock can be released with /DRIVEOFF only (the MO disk can be changed after a waiting time of 20s max.).

Thus to change the MO disk, you must always execute the two commands /DRIVEOFF and /DRIVEON.
(Although the /DRIVEON function is often skipped by the ORG-M user, it is nevertheless the correct procedure for changing a data medium as specified in the ORG-M system description.)

:INIT:FORMAT...

The formatting command should always include the product identification 'P20PSD' and the additional instruction FIRST:

```
:INITM:FORMAT PLSKxx-P20PSD FIRST
```

All other formatting commands format the entire MO disk resulting in the loss of all data.

:INIT:INIT...

The initialization command should always include the parameter LS (of the same size as defined for structuring).

```
:INITM:INIT PLSKxx-P20PSD LS-xxxxx
```

If LS is not specified, INITM might respond with the message 'Block factor > 1'.

:INITM:DELETE...

M2000 does not support the deletion of logic sub-devices. The function is executed without effect and indication.

Neither does M2000 implement the DEL-PTR function (delete pointer).

The other INITM functions however, such as initialization after deletion (not the parameter NOINIT), are executed.

6.15.2 SICOMP R Computer

The SICOMP R emulation SIC-R replaces the SICMP R minicomputer in nearly all aspects. Although special emphasis has been placed on compatibility to the original system, some restrictions in regard to hardware compatibility must be accepted whose remedy would be too costly and in some cases even are desirable (see "time behavior" below).

These restrictions are described in the following.

Floating point values

During the transfer of floating point values from an emulated ORG system to the Windows system (usually via PSD/PSAPI), the mantissa are processed differently.
Original SICOMP K format: 56-bit mantissa
Microsoft DOUBLE format: 53-bit mantissa

Time behavior

Emulations based on today's modern PC hardware offer much more performance than the original SICOMP M with its out-dated components, with the result that SICOMP R application systems run much faster. Disk access in particular can be speeded up by factor 50 to 100 due to the high cache hit rates.
This "incompatibility" in the time behavior is a desirable side effect of an emulation (a well designed application software usually handles this increase in performance without problems).

Timer

Due to the use of standard hardware components and the Windows operating system, limitations with regard to time of day granuality and shortest possible cyclic wake-up time must be taken into account for time-modified calls. The shortest possible time unit is 10ms (resolution).

Floppy disk

The data structure of a floppy disks processed by the emulation corresponds to that of the original system. However, data exchange via this medium is not possible as today's floppy disk controllers no longer support the specific modulation of track 0.

DUST3966

The DUST3966 device emulation does not support the WIKOP functions (Remote loading ...).

6.16 Communication Channel SCU – SICOMP M/R Programs

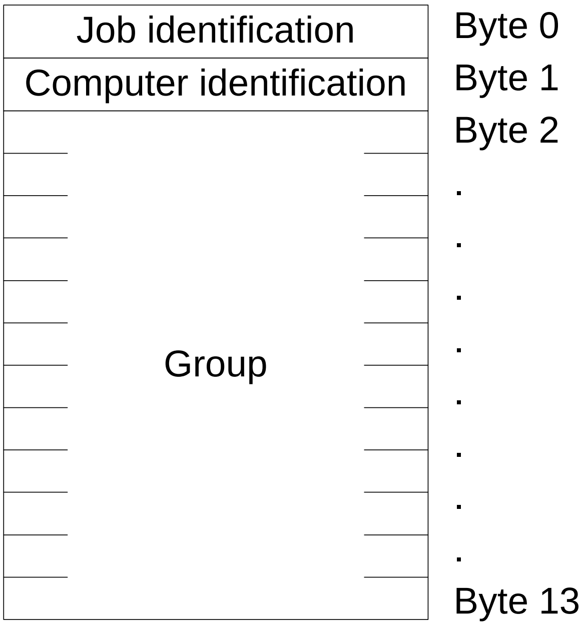
```
DEVICE = ioadr,PSCU
DEVICE = anr,gnr,PSCU
```

ioadr IO address according to ORG generation
(4-digit hexadecimal number).

anr,gnr Connection point number/device number according to ORG generation (decimal number).

M2000 sets up a communication channel to the SCU service and connects it to the DSSK display unit channel generated in ORG. The user program exchanges messages with the SCU service over this display terminal channel (ORG calls $STAUAL/$STEIAL).
The messages are not explicitly acknowledged; a receive call ($STEIAL) must always be pending in ORG-M/R.

Structure of the user messages:



Job identification

SCU → User program
0 Switching operation is executed;
the following bytes contain the actual information
1 Status information;
the following bytes contain the actual information

User program → SCU
10 Initiate switching operation;
Byte 1 is empty, bytes 2 to 13 contain the name of the switching group
11 Current status request;
bytes 1 to 13 are empty

Computer identification

Identification of the process controlling computer (0...9) acc. to SCU parameter definition.

Group

Name of the group switched last acc. to SCU parameter definition.

7 Auxiliary Programs

7.1 The PCCOPY Program

The PCCOPY program can be used to export library elements or ORG files from the SICOMP M/R system to the Windows file system, and to import Windows files:

- Export: SICOMP M/R library elements or files are stored as Windows files in a Windows directory.
- Import: A Windows file is stored as file or library element in the SICOMP M/R file system.

PCCOPY can be run in the SICOMP M/R system.

To install PCCOPY under AMBOSS/BS-M, follow the procedure below:

- In the parameter file (*mpar.sys*), specify the pathname of one of the generated data disks so that it points to the file *plsk.psd*.

  (part of the M2000 package, available after installation of the user system).

Example:

```
device = 8400-0, FP0XX,1995, "..\disks\plsk.psd";
device = 8,0, 3949, "..\disks\plsk.psd";
```

- Start the emulator.
- PCCOPY is located in the GSB library on the data disk and can be loaded as PRP or HRP.
- After the program has been loaded into the system, the parameter file can be reset to the previous state.
- To view the functions of PCCOPY, enter :PCCOPY:?.

```
:PCCOPY:?
COPY  COPY FILES AND LIBRARIES TO AND FROM DOS PARTITION
COPY device-lib.[element] TO dospath   Copy Library Element(s) to DOS-File(s)
COPY device-filename TO dospath        Copy File(s) to DOS-File(s)
COPY dospath TO device-lib.[element]   Copy DOS-File(s) to Library
COPY dospath TO device-filename        Copy DOS-File(s) to File(s)

Examples:
COPY PLSK1-QSB. TO C:\MYDIR            Copy all Library Elements to Directory MYDIR
COPY PLSK1-QSB.A* TO C:\MYDIR          Copy all Library Elements A* to Dir. MYDIR
```

Notes

PCCOPY uses calls for reading and writing data in ORG BIBEAS. As PCCOPY itself does not convert characters (exception: see section d)), the following must be taken into account when copying data from Windows to ORG:

a) The source element contains vowel mutation (umlaut) such as ä (E4), Ä (C4), ü (FC), etc. corresponding to the coding of the character table used in Windows. BIBEAS writes these characters to the target file according to the source element. As this character coding cannot be represented with the MEDIS editor, the vowel mutation must be written as 'ae' (ä), 'ue' (ü) in line with ORG conventions.

b) The source element contains horizontal tabs. The tabs are correctly stored with 09 in the target element during copying. If in ORG the target element is read in 'unpacked' mode via BIBEAS, FILE / MEDIS, the horizontal tabs are replaced by the corresponding number of blanks according to BIBEAS rules so that a MEDIS line could then be longer than 80 characters. Horizontal tabs HT can only be correctly interpreted by BIBEAS if they are stored according to BIBEAS rules. These problems do not arise if you use the 'packed' mode for the reading in process. To generally avoid these problems, horizontal tabs in the source element should be replaced by blanks in Windows.

   When data are copied from ORG to Windows, PCCOPY tries to create a file with the element name according to the DOS conventions 8+3. Blank characters in the file name are not allowed.

c) When data are copied from ORG to Windows, ETX is copied to line feed (0A). The copied files can then be displayed and edited using a Windows text editor. When files are copied back from Windows to ORG, line feed (0A) is reconverted to ETX.

d) If the last character of the target name is a $ character when copying data from ORG to Windows, the data will be stored under this name.
   If the $ character is not included, the source(file/BibElement)name is used as the target name.

Example *:PCCOPY:COPY PLSK-BIB.ELNAME TO C:\ZDIR*
The data is stored under ELNAME in ZDIR.

:PCCOPY:COPY PLSK-BIB.ELNAME TO C:\ZDIR\ZNAME.TXT$
The data is stored in ZNAME.TXT located in ZDIR.

7.2 The MCSAVE.EXE Program

MCSAVE is a Windows program that can be used for importing MK82 tapes to the Windows system. To view the available parameters of the *mcsave.exe* program, enter the command 'mcsave -?' (see below).

usage:	mcsave [-T geraet] [-van][-pxx][-r][-d]
-T	Device specification in WINDOWS notation
-v	Volume name is used as file name, otherwise plskxx
-pxx	First file name plskxx, xx=00...09, (otherwise plsk00)
-a	Read in all volumes of the magnetic tape
-n	Do not rewind tape
-r	Rewind tape only
-d	Read only one (the first) volume label
-D	Read all volume labels

Note With parameter -d or -D, the program reads the first (-d) or all labels (-D), but does not copy any data.

Note The functions of the *mcsave.exe* program are available on the M2000 tab `General – Import data.`

7.3 The WAIT4.EXE Program

The WAIT4.EXE program waits until Windows objects can be accessed or an M2000 status is received.

The WAIT4.EXE program can be activated via the M2000 start batch. It then waits until all Window drivers and services have been started before M2000 is started.

The WAIT4.EXE program can also be activated via user-specific start batches. It then waits until specific pipes, for example, are created by M2000.

Program activation possibilities and examples:

wait4 com99 The name com99 is extended to \\.\com99. The program waits until \\.\com99 can be accessed.

wait4 \\.\pipe\coil2in The name \\.\.... is accepted without changes. The program waits until pipe *coil2in* can be accessed.

wait4 file.1 -n Parameter -n forces the program to accept the name *file.1* without changes (compare to *wait4 com99*). The program waits until *file.1* can be accessed.

To view the available program parameters, enter *wait4 -?* or *wait4 /?*:

```
WAIT4 [name] [-tzz][-xzz][-n][-w][-r][-d][-?]
    name: name of the NT devices to wait for default: NT service database
    -tzz: wait Timeout (zz sec),      default = 60
    -xzz:delay zz sec before exit,    default = 2
    -n:   No name conversion          (*)
    -r:   test with Read access       (*)
    -w:   test with Write access      (*)
    -d:   Debug mode
    -?:   this screen                 (*) d.c by NT service database:
```

7.4 The SOLDIS Program

The Windows program SOLDIS can read data (including data written on SICOMP M/R) from an MO disk and write this data to Windows. It can also be used for writing Windows files to an MO disk.

Call:
soldis [options] file1 file2 ...

options:
 /s
Write data from file1, file2 ... to the MO disk (default setting)
 /l
Write data from the MO disk to file1, file 2 ...
 /d#
Specification of the physical drive number # (1..n). This entry is usually not required as SOLDIS determines the drive numbers automatically. It is required however, if several MO drives are connected to the computer.
 /type
Specification of the emulated disk type:
 /3941
 /3942 (default setting)
 /3945
 /3948a
 /3949
 /n
Do not generate start message
 /v
verbose: generate runtime messages for error correction

7.5 SicView, M2000 Hard Disk Explorer

The emulator represents the hard disks of original SICOMP systems by means of WINDOWS files. Each one of these files contains a binary image of an original hard disk. Access to the data of such a hard disk image is normally only possible via the emulated SICOMP system.
Users with only little knowledge about ORG-M/R usually have some difficulties in handling the emulated SICOMP system. **SicView** allows these users to view SICOMP data in the familiar Windows way.

SicView allows the user to visualize the contents of a SICOMP disk and the disk journaling system (also via network). This way SICOMP data can also be accessed by users that are not quite so familiar with the SICOMP system. **SicView** reads the binary image of a SICOMP disk and displays the structure in a manner similar to the Windows Explorer.

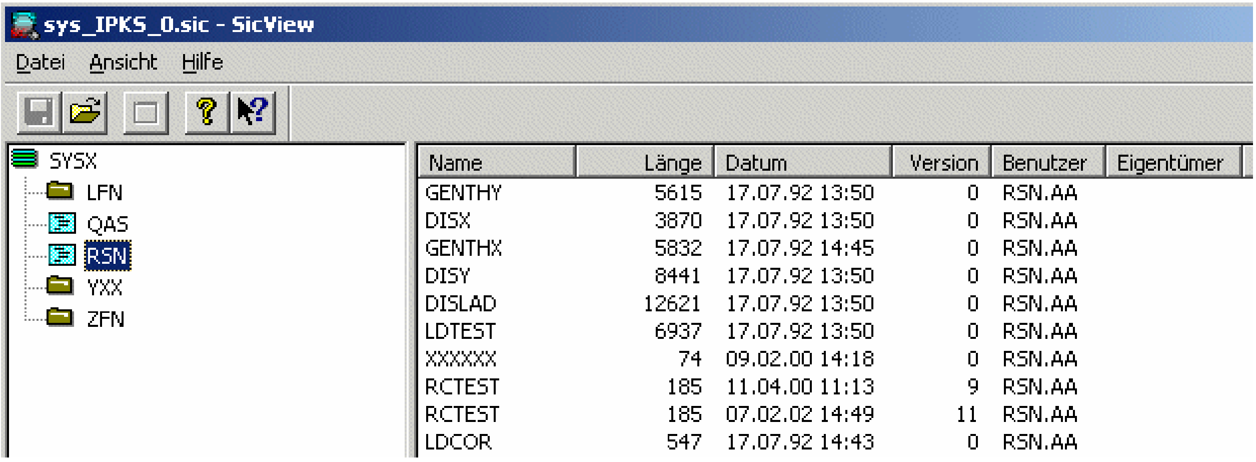
Program features

- Representation of the SICOMP disk contents
- Display of ORG files and library elements in HEX, ASCII and WORD representation
- Printout and preview of ORG files and library elements
- Export of ORG files and library elements to normal Windows files. Import of changed QS elements/files, but not of GS elements/files. (Export/import with PCCOPY only).

As **SicView** only reads the (virtual) SICOMP disk in offline mode, it is not possible to change files with the **SicView** program.

In the parameter file *mpar.sys/rpar.sys*, the parameter *mode=share* must be specified to avoid problems when accessing an exclusively occupied volume.

It is possible to automate **SicView** functionality. The M2000 package contains an example illustrating how SICOMP data can be accessed from an Excel table via *Visual Basic for Applications* (VBA).
SicView is part of the M2000 package and must be enabled in the dongle. To start **SicView**, open the `General` tab of M2000 and click the **SicView** button.
.



Name	Länge	Datum	Version	Benutzer	Eigentümer
GENTHY	5615	17.07.92 13:50	0	RSN.AA	
DISX	3870	17.07.92 13:50	0	RSN.AA	
GENTHX	5832	17.07.92 14:45	0	RSN.AA	
DISY	8441	17.07.92 13:50	0	RSN.AA	
DISLAD	12621	17.07.92 13:50	0	RSN.AA	
LDTEST	6937	17.07.92 13:50	0	RSN.AA	
XXXXXX	74	09.02.00 14:18	0	RSN.AA	
RCTEST	185	11.04.00 11:13	9	RSN.AA	
RCTEST	185	07.02.02 14:49	11	RSN.AA	
LDCOR	547	17.07.92 14:43	0	RSN.AA	

8 PE F7 Emulation

8.1 Hardware Structure

8.1.1 SICOMP M

The SICOMP M system is capable of operating digital and analog I/O modules in up to four EA 04 subracks, each subrack providing 19 free slots.
Up to four additional EG 180 or EG 182 expansion units (with 8 or 6 expansion slots each) can be operated by installing an EG-AS 300-S interface module in one of these slots.
The system can thus be expanded to a maximum configuration of 4 * (18 + (3 * 8)) = 168 modules. Based on the maximum number of 32 inputs or outputs per module, the system would then be capable of processing 5376 inputs/outputs or 672 I/O bytes.

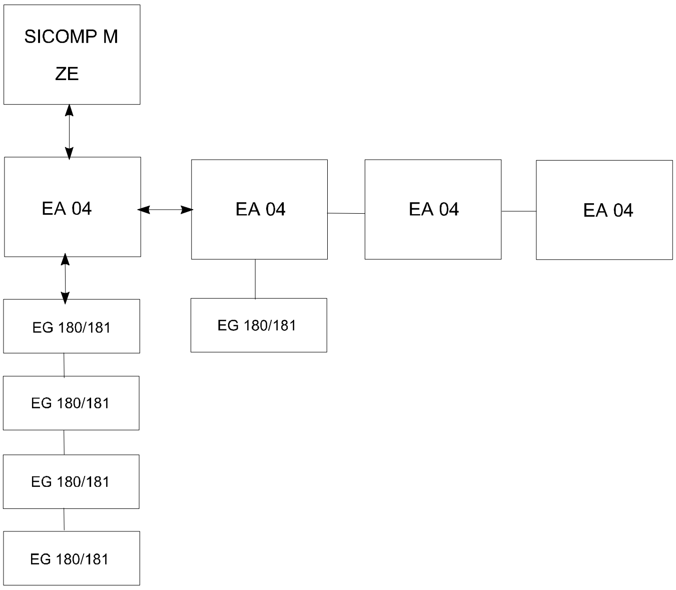


Figure 6: Process computer peripherals

8.1.2 M2000

The SICOMP M configuration shown in the figure above is simulated in the M2000 system using the Siemens Profibus as interface to the process I/O equipment.
In the emulation, S7 modules implement the connection to the process I/O equipment - analogous to the EA04 subracks for the SICOMP-M system. A CP 5611 Profibus module is installed in the M2000 computer connecting Windows/M2000 to the S7 hardware.

The resulting configuration is shown in the below diagram:

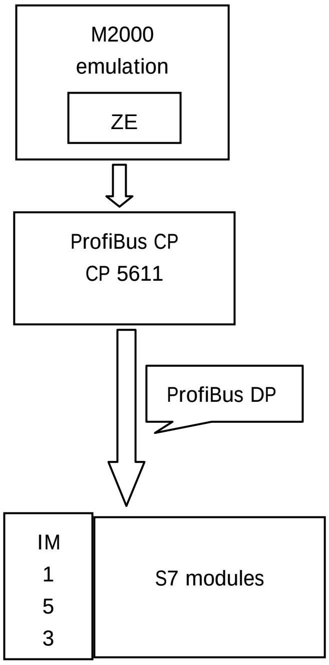


Figure 7: M2000 process I/O equipment

8.2 Software Structure

The figure below illustrates how SICOMP M user programs running under M2000 access the process I/O equipment (Profibus).

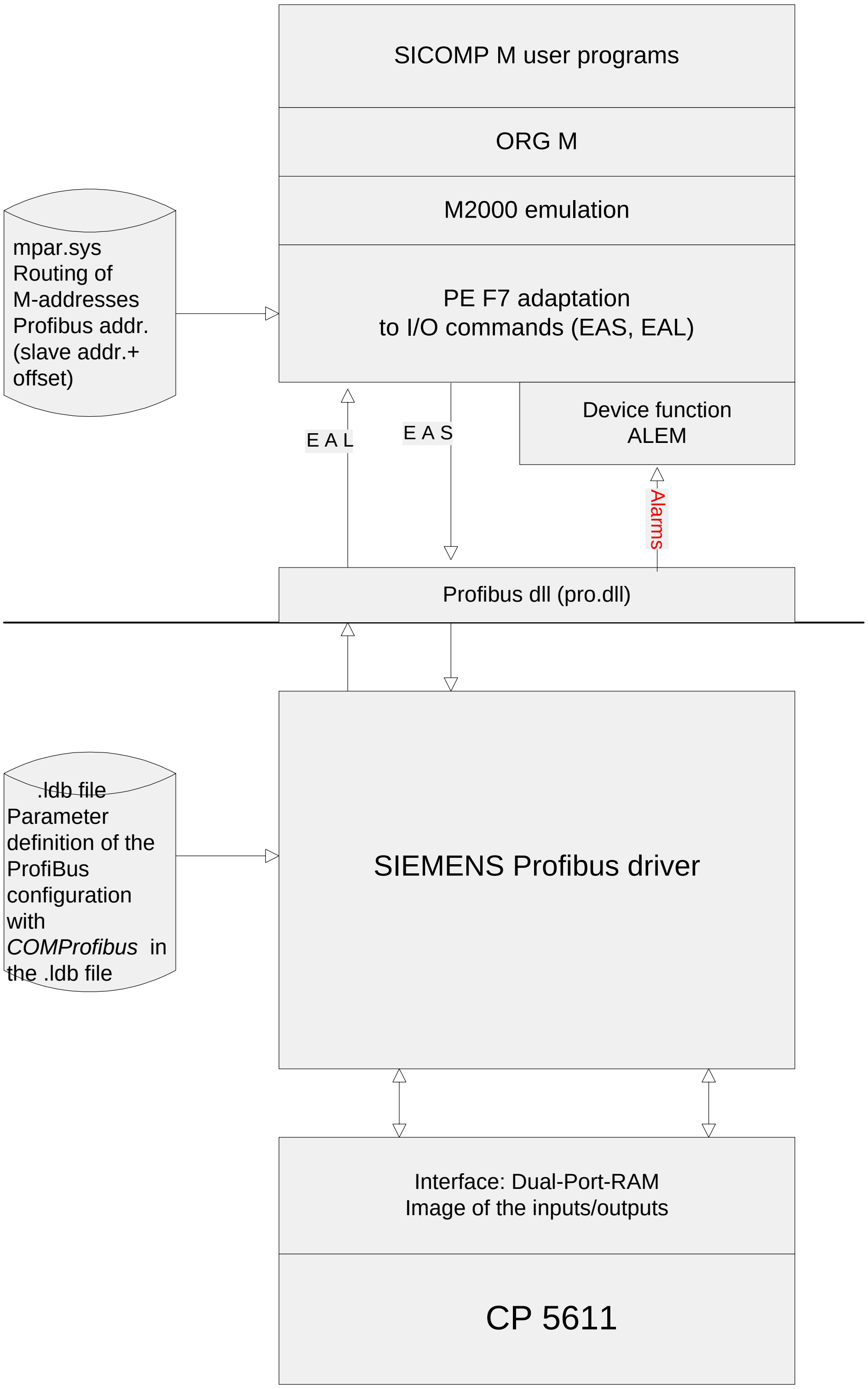


Figure 9: Access via Profibus

8.2.1 Access to the Process I/O Equipment

M2000 provides a **PE F7** device emulation for the I/O access to the process peripherals. This device emulation directly processes the process calls that are not allocated to a process device generated in ORG-M. Process calls using the process device ALEM are processed by an M2000 **ALEM** device function. Output commands to the process peripheral equipment are transmitted by ORG-M in the form of EAS commands to the PE F7 peripherals. In the M2000 device emulation, these commands are forwarded to the ProfiBus-DLL (*pro.dll*), which in turn passes them on to the ProfiBus driver. This way the output data are transmitted to the dual-port-RAM of the CP 5611 module. The CP 5611 module cyclically compares the state of the DPR to that of the hardware. In the opposite direction, following the same procedure, inputs from the DPR of the CP are made available to the EAL commands of the ORG system.

8.2.2 Alarm Processing

As ProfiBus does not alert the user programs about data changes, the M2000 ProfiBus-DLL takes over this function by reading out the relevant input data at defined cycles and alerting the ORG systems when data changes are detected.

8.3 Functional Description

8.3.1 Device Emulation

In M2000, the PE F7 device emulation processes all calls to the process peripheral equipment. These calls to the process peripheral equipment are divided into element calls (calls directly processed in a user program run) and calls that access a process device generated in ORG-M.
Element calls (EAL/EAS) are processed directly by the PE F7 device emulation. The calls to a process device are passed on by the PE F7 device emulation to a device function (thread) for processing.

8.3.1.1 Element Calls

The ORG-M element calls, such as BYTEINR, BYTAUSR, etc., are implemented as subprograms in the PE F7 adaptation.

Upon detecting such a command, the M2000 ZE emulation activates a subprogram in the *pro.dll* that converts READ and WRITE calls directly to the corresponding Siemens ProfiBus instructions.

Note In the SICOMP M system, these calls are prioritized in the PE F7 interface and cannot be interrupted by other events (change state interlock is set).

In the M2000 parameter *mpar.sys*, the PSF address used in ORG-M is converted to a ProfiBus address. The program status register (PZR) is set according to the transfer results of the I/O command.

8.3.1.2 Device Function

The process devices generated in ORG-M are processed by means of a device function (thread). This device function processes ORG-M calls such as ALANM to device ALEM, etc.

Upon detecting an EAL command for a device ALEM, the M2000 ZE emulation activates the device function, which then transmits calls to the ProfiBus-DLL (*pro.dll*) and returns call results and alarm events to the central unit ZE.

In the M2000 parameter *mpar.sys*, the PSF address used in ORG-M is converted to a driver data address (ProfiBus address). The ProfiBus address and values (8-bit) are transmitted to the ProfiBus-DLL by means of the corresponding read or write instruction. The return value is converted into the ORG call indication according to the ORG-M specifications.

8.3.1.3 Addressing

The address consists of the corresponding ProfiBus slave number and an offset value.

Example Address:$_{\text{(hexa)}}$

3 0 0 0 4 Addressed is byte no. 4 in the address space of slave no. 3.

Slave no. Offset(4-digit)

8.4 ProfiBus DLL

8.4.1 General

The M2000 ProfiBus DLL has the following tasks:

- Initialization of the ProfiBus interface upon emulation start
- Conversion of read/write instructions into ProfiBus calls
- Generation of alarms upon changes in the relevant input data

8.4.2 Installation (ProfiBus)

Please refer to the
description in the Siemens SIMATIC-NET package.

Note

Install the ProfiBus following the procedure on the current SIMATIC-NET CD-ROM and define the parameters for the used configuration.
M2000 uses the CP_L2_1 interface for access to the ProfiBus.

8.5 Parameter Definition

8.5.1 Configuration File *mpar.sys*

The process parameters are defined centrally by means of the M2000 configuration file *mpar.sys*. This file must contain the entries described below.

A ‘device’ entry to integrate the PE_F7 calls in the emulation.

DEVICE = *ioadr*, ALEM, "PROFIBUS"

A routing entry for every peripheral byte showing the address allocation and the type and parameter definition of the signal converter.

Profibus **PSF**=*sicAdr*,pro*adr*,*typ*,[*StaDyn*][*Flank*][,*Inter*][,*Analogm*]

Description of the instruction parameters:

sicadr Address offset of the module relative to *ioadr*

proadr Associated Profibus address, hexadecimal, in the format xxxxyyyy; xxxx denotes the slave number, yyyy the address of the module.

typ Type of SICOMP-M module according to the syntax of the SICOMP-M test program TESIG:

DE8M 8 digital inputs, occupy 1 byte in the SICOMP-M address space
DE16M 16 digital inputs, occupy 2 bytes in the SIC-M address space
DA8M 8 digital outputs, occupy 1 byte in the SIC-M address space
DA16M 16 digital outputs, occupy 2 bytes in the SIC-M address space
AE8M 8 analog inputs, occupy 16 bytes in the SIC-M address space, left justified representation
AEPB 8 analog inputs, occupy 16 bytes in the SIC-M address space, right justified representation
AA8M 8 analog outputs, occupy 16 bytes in the SIC-M address space, left justified representation

StaDyn Module type: STATIC or DYNAMIC, must be specified for digital input groups (DE8M or DE16M).

Flank Bit mask for the active edge of dynamic modules (only StaDyn = DYNAMIC)
0: input responds to negative edge
1: input responds to positive edge

Inter For digital input modules (DE8M or DE16M), the assigned SICOMP-M interrupt must be specified if the modules generate alarms.

Analogm For analog modules, representation mode conversion can be specified. If analog mode is not specified, the values are transmitted in 'complement on one' format.

2ER_KOMPLEMENT: The values are transmitted in 'complement on two' format (analog input only).
S7TRANS: The values are transmitted in S7 format. The format is not adapted..
4_20MA: Format adaptation between 4-20mA and S7/S5 representation.

8.5.2 Example (ProfiBus)

The parameter file *mpar.sys* contains the following instructions:

```
device = 2000,ALEM,“PROFIBUS“
psf = 0100,30010,AEPB,2ER_KOMPLEMENT          1.)
psf = 0040,40000,DA16M                        2.)
psf = 0020,40002,DE16M,STATISCH               3.)
psf = 0022,40002,DE8M,DYNAMISCH,F0            4.)
```

1. 8 right justified analog inputs (AEPB) are set under DeviceAdr 2100_{16} to $210F_{16}$. The values are transmitted in 'complement on two' format generated from the input byte 16-31 (3**0010**)$_{16}$ in ProfiBus slave no. 3 (**3**0010)$_{16}$.
2. 16 digital outputs are set under DeviceAdr 2040_{16}. The values are generated in slave no. 4, byte 0 and byte 1.
3. 16 digital inputs are set under DeviceAdr 2020_{16}. The values are generated in slave no.4, byte 0 and byte 1. Module type = static.
4. 8 digital inputs are set under DeviceAdr 2022_{16}. The values are generated in slave no.4, byte 2. Module type = dynamic. Inputs 0 to 3 respond to negative edge, inputs 4 to 7 to positive edge.

Note Dynamic inputs are simulated by the emulation. The S7 contains static standard input modules only.

9 Appendix

9.1 General Notes

Floppy disk FLOP

A floppy disk used in the emulated SICOMP system must be formatted with the corresponding SICOMP system functions.
A floppy disk in DOS format can be used in the emulated system if it is processed with INIT. After having been processed with INIT, the floppy disk can no longer be read in DOS. The FLOP disk device can be used for the data exchange between different emulated SICOMP systems using 5¼“ or 3½“ floppy disks. These floppy disks cannot be read in the original SICOMP system.

FACO30

M2000 allows you to send hardcopy outputs to a connected EPSON compatible printer (the EPSON printer must be specified in FACO30).

Printers

It is basically possible to connect any printer to the computer's serial or parallel interfaces. However, the serial interface should be preferred as the Windows printer for the parallel interface implements polling, which might have negative effects on the response time behavior of Windows.
Only printers operated in ASCII mode by default and with their own character set should be employed.

Printer control characters

Control characters to the printer (e.g., for typeface switching) are not modified by M2000. It may be necessary to adapt the application programs if it is not possible to use the same or a compatible printer in connection with the M2000 emulator.

Dongle

M2000 requires the use of a dongle connected to the parallel printer port.

Dongle label information

M2000	Program name
M2-xxx	Name of the basic package
+ Optionen	Optional packages
Dongle-Nr.:	
xxxxx	Five-digit dongle number

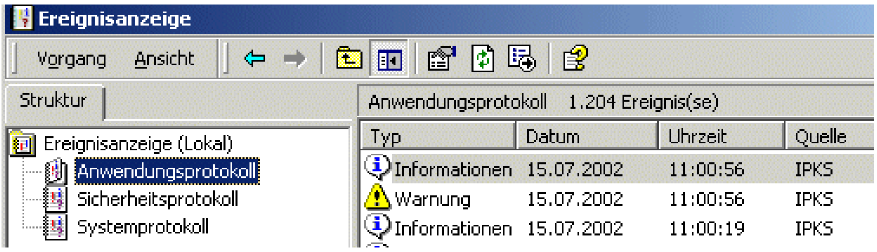
DFCONS

Other than the SICOMP system, the computer does not provide a buffered RAM memory. By using the *IMAGE* parameter, the main memory contents is saved upon termination of the emulation and preallocated upon emulation restart. Thus DVS files or DVS journaling data can be reconstructed with DFCONS after resumption of the power supply.

9.2 Messages

9.2.1 Start, End and Error Messages

M2000 writes its start and end messages, warnings and error messages to the Windows event log (application log). These messages have the same structure as the Windows messages. M2000 messages can be identified by the 'IPKS' designation in the 'Source' column.

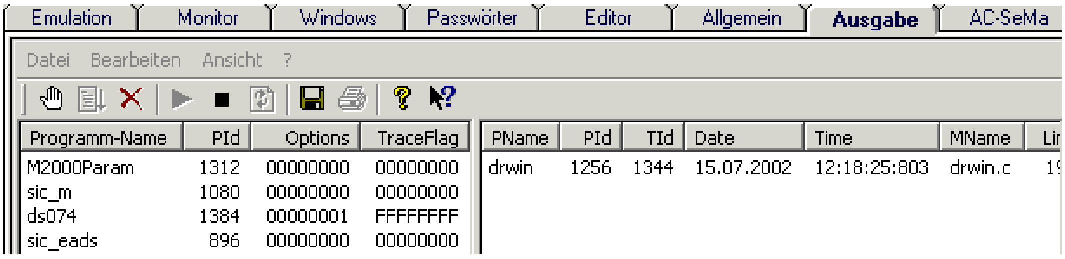


Anwendungsprotokoll 1.204 Ereignis(se)

Typ	Datum	Uhrzeit	Quelle
Informationen	15.07.2002	11:00:56	IPKS
Warnung	15.07.2002	11:00:56	IPKS
Informationen	15.07.2002	11:00:19	IPKS

9.2.2 Trace Messages

9.2.2.1 TrcView Window



Programm-Name	PId	Options	TraceFlag
M2000Param	1312	00000000	00000000
sic_m	1080	00000000	00000000
ds074	1384	00000001	FFFFFFFF
sic_eads	896	00000000	00000000

PName	PId	TId	Date	Time	MName	Lir
drwin	1256	1344	15.07.2002	12:18:25:803	drwin.c	19

In the TrcView window, the pane to the left displays the Process list, the pane to the right the Trace list. The process list contains all processes that are linked to the TRC interface (*trcdll.dll*).
By double clicking an entry in the process list, the options dialog for the selected process opens and the trace message output can be defined. See also:
`TrcView Application Help.`
The associated trace messages are displayed in the Trace list.

9.2.2.2 TrcView Features

The TRC module offers an easy-to-handle and user-friendly standard trace and error signaling interface under Windows. The trace messages are recorded in a main memory cyclic buffer to minimize the effects on the system's time behavior.

The trace interface offers the following features:

- Message header with process name, process ID, thread ID, date/time stamp, module name, source line number
- Trace of a buffer in ASCII and hexadecimal format
- Output type selection (cyclic buffer and/or file) at runtime
- Message output in cyclic buffer minimizes effect on time behavior
- Adjustable message size and cyclic buffer size
- Settings can be saved in the recording system (Save button)
- User-friendly indication system with list of all processes with trace output, for message activation/deactivation and cyclic buffer display.

9.3 Template of the Parameter File *mpar.sys*

```
;-----------------------------------------------------------------------------
;-----------------------------------------------------------------------------
; M2000: M-emulation: parameter definition example, formatted 80 characters/line
; The file can also be edited with MEDIS
;-----------------------------------------------------------------------------
; no vowel mutation, MEDIS compatible !!!
;-----------------------------------------------------------------------------
; Basic parameters
;-----------------------------------------------------------------------------
; cpu = ze01                    ; Type of the central unit ZE
; cpu = ze02
cpu = ze03
; zestart = 7                   ; ZE start delay = 7 seconds
; mode=popup                    ; PopUp window indicating internal errors
maxmem = 4032                   ; Main memory size in kwords
; image = "image"                ; HSP retrieval file
BOOT = 8200,8008
;       |    |______________IO address of the virtual console
;       |___________________IO address of the BOOT disk
;
;------- if subaddress of the boot disk, see readme.wri !! --------
;
; console in local window
;-----------------------------------------------------------------------------
device = 8008,DS074,"DSSK0"
;-----------------------------------------------------------------------------
; DISIT screen in local window
;-----------------------------------------------------------------------------
; device = 8008,3974MT,"DSSK0"
;-----------------------------------------------------------------------------
; console without color option on serial interface
; (parameterize COMx with setall.bat)
;-----------------------------------------------------------------------------
;device = 8008,DS075,"\\.\COM2"
;  own dongle available
;device = 8008,DS075,"\\.\COM2",IPKS
;  central dongle available
;device = 8008,DS075,"\\.\COM2",201,NONE,500
;  inoperable detection on, for V.24 interface
;  overparameterization not possible
;  own dongle available
;  delay time 500 ms
;-----------------------------------------------------------------------------
; console with color option on serial interface
;(parameterize COMx with setall.bat)
;-----------------------------------------------------------------------------
;device = 8008,DS075F,"\\.\COM2"
;  own dongle available
;device = 8008,DS075F,"\\.\COM2",IPKS
;  central dongle available
;-----------------------------------------------------------------------------
; DISIT screen on serial interface
; (parameterize COMx with setall.bat)
;-----------------------------------------------------------------------------
;device = 8030,3974MTSER,"\\.\COM2"
;  overparameterization not possible
;  own dongle available
;device = 8030,3974MTSER,"\\.\COM2",1
;  overparameterization not possible
;  own dongle available
;device = 8030,3974MTSER,"\\.\COM2",1,IPKS
;  overparameterization not possible
;  central dongle available
;device = 8030,3974MTSER,"\\.\COM2",0
;  overparameterization possible
;  own dongle available
;device = 8030,3974MTSER,"\\.\COM2",0,IPKS
;  overparameterization possible
;  central dongle available
;--------------------------------------------------------------------------
; VDU2000 on serial interface
; (parameterize COMx interfaces with setall.bat)
;--------------------------------------------------------------------------
;device = 8038,VDU2000,"\\.\COM2"
;  overparameterization not possible
;  own dongle available
;device = 8038,VDU2000,"\\.\COM2",1
;  overparameterization not possible
;  own dongle available
;device = 8038,VDU2000,"\\.\COM2",1,IPKS
;  overparameterization not possible
;  central dongle available
;device = 8038,VDU2000,"\\.\COM2",0
;  overparameterization possible
;  own dongle available
;device = 8038,VDU2000,"\\.\COM2",0,IPKS
;  overparameterization possible
;  central dongle available
;-----------------------------------------------------------------------------------
; DSSK via LAN, more than one NetCard installed in PC
; (network software (TCP/IP) must be installed in Windows)
;------------------------------------------------------------------------------------
;DEVICE=8030,3974MTNETF,192.168.2.10:20010/0,IPKS
;DEVICE=8030,3974MTNETF,NetCard-LAN1:20010,IPKS
;------------------------------------------------------------------------------------
; console via SOCKET
; (network software (TCP/IP) must be installed in Windows)
;-----------------------------------------------------------------------------------
;device = 8008,DS074NET,22220
;  own dongle available
;device = 8008,DS074NET,22220,IPKS
;  central dongle available
;----------------------------------------------------------------------------------
; DISIT screen via SOCKET
; (network software (TCP/IP) must be installed in Windows)
;----------------------------------------------------------------------------------
;device = 8040,3974MTNET,22220
;  own dongle available
;device = 8040,3974MTNET,22220,IPKS
;  central dongle available
;---------------------------------------------------------------------------------
; DS078 screen (TE2078 emulation) via SOCKET
; (network software (TCP/IP) must be installed in Windows)
;---------------------------------------------------------------------------------
; device = 8410, DU78, "SOCKET", 20000,192.168.1.49:20001;
;---------------------------------------------------------------------------------
; hard disk drives
; Attention ! Hard disk entries must be sorted by device number DevNr
; in ascending order
;
```

```
; optional read access to the disks from other programs
; (copy, SICVIEW, ...)
;MODE = SHARE
;---------------------------------------------------------------------------
; hard disk drive PS048,PS049
;---------------------------------------------------------------------------
;device = 8200-0,PS048,"..\platten\plsk00"
;  64640 sectors / without WRITE-THROUGH-Flag
;device = 8400-0,PS049,"..\platten\plsk10",1
;  25856 sectors / with WRITE-THROUGH-Flag
;---------------------------------------------------------------------------
; hard disk drive FP0xx
;---------------------------------------------------------------------------
;device = 8100-0,FP0XX,64640,"..\platten\plsk00"
;  64640 sectors / without WRITE-THROUGH-Flag
;device = 8200-0,FP0XX,25856,"..\platten\plsk10",1
;  25856 sectors / with WRITE-THROUGH-Flag
;device = 8400-0,FP0XXAE,64640,"..\platten\plsk00"
;  64640 sectors / without WRITE-THROUGH-Flag
; with ASCII-EBCDIC conversion
;---------------------------------------------------------------------------
; floppy disk drive
;---------------------------------------------------------------------------
;device = 8104,FLOP,"\\.\A:"
;  floppy disk drive a:        15 sectors format
;device = 8104,FLOP18,"\\.\B:"
;  floppy disk drive  b:       18 sectors format
;---------------------------------------------------------------------------
; magnetic tape cassette drive MK82
;  Tandberg drive driver must be installed in Windows
; drive access in Windows can be tested with the function
; administration->tape drive.
;---------------------------------------------------------------------------
device = 8106,MK082,"\\.\TAPE0"
;  "first" drive acc. to numbering in Windows
;device = 8106,MK082,"\\.\TAPE1"
;  "second" drive acc. to numbering in Windows
;---------------------------------------------------------------------------
; optical disk OP31
; only one entry for every drive connected to the PC
; (there are no subdevices)
;---------------------------------------------------------------------------
;device = 8106,OP31,"\\.\PhysicalDrive0"
;  "first" drive acc. to numbering in Windows
;device = 8106,OP31,"\\.\PhysicalDrive1"
;  "second" drive acc. to numbering in Windows
;---------------------------------------------------------------------------
; printers
; (parameterize COMx interfaces with setall.bat)
;---------------------------------------------------------------------------
;device = 8020,DRUA,"LPT1"
;  printer on parallel port
;  write monitoring time 20s
;device = 8020,DRUA,"LPT1",20
;  printer on parallel port
;  write monitoring time 20s
;device = 8020,DRSPOOL,"LPT1"
;  printer on parallel port
;  operation via WINDOWS driver
;device = 8090,DRSPOT,"I:\DRFILE\DRDATEN",30
;  print to file, not DOS compatible
;  time-controlled closing of file
;device = 8090,DRSPOTS,"I:\DRFILE\DRDATEN",30
;  print to file, not DOS compatible
;  time-controlled closing of file
;device = 8090,DRSPOTX,"\sicomp\spotx.par"
;  print to file, parameter-controlled
;device = 8020,DRUA,"..\Drucker\drua0.txt"
;  printer output to WINDOWS file
;device = 8020,DRWIN,"drua0"
;  printer output to WINDOWS window
;device = 8020,DR202,"\\.\COM7"
;  DR202 printer on serial port
;  write monitoring time 20s (default)
;  no break detection
;device = 8020,DR202,"\\.\COM7",20,1
;  DR202 printer on serial port
;  write monitoring time 20s
;  break detection
;device = 8020,DRCOM,"\\.\COM7",8,30,100
;  printer on serial port
;  do not parameterize serial port, mode command applies
;  write monitoring time 30s
;  waiting time 100 ms
;---------------------------------------------------------------------------
;  DU03 data transmission control
;---------------------------------------------------------------------------
; device = A2C8,DU03,"NDIS",0;           for WindowsNT
; device = A2C8,DU03,"NDIS2000",0;    for Windows2000
;---------------------------------------------------------------------------
; DU04 computer interfacing
; (parameterize COMx interfaces with setall.bat)
;---------------------------------------------------------------------------
;device = A2C0,DU04,"\\.\COM2",11
;  via serial interface
;  high priority / with BCC
;device = A2C0,DU04,"\\.\COM2",00
;  via serial interface
;  low priority / without BCC
;device = A2C0,DU04,"\\.\DFXX0",00
;  via COMSOFT DF32/DF42
;  board 0 / interface 0
;device = A2C0,DU04,"\\.\DFXX0",12
;  via COMSOFT DF32/DF42
;  board 1 / interface 2
;---------------------------------------------------------------------------
; DU05 computer interfacing
;---------------------------------------------------------------------------
;device=8300,DU05,"\\.\DFXX0",512,g:\sicomp\du05.par;
; via COMSOFT DF42
;device=8300,DU05,"SOCKET","WINNT1",22220,512,20000,10000,30000;
; gateway function
; active side at connection buildup, enter host name
;device=8300,DU05,"SOCKET","",22221,512,20000,10000,30000;
; passive side at connection buildup, empty string
;device=B400,DU05,"SOCKET","partner-11.12.13.1",20000,0;
; two network cards
;device=B400,DU05,"SOCKET","partner-LAN1",20000,0;
; connection established via network card 1
;device=B400,DU05,"SOCKET","partner-11.12.13.2",20000,0;
;device=B400,DU05,"SOCKET","-11.12.13.2",20000,0;
;device=B400,DU05,"SOCKET","partner-LAN2",20000,0;
```

```
; connection established via network card 2
;----------------------------------------------------------------------------
; DU06 computer interfacing
;----------------------------------------------------------------------------
;device=A2C0,DU06,"\\.\DFXX0",0,0,UP,1,512,0,0,0,du06up.par
;  via COMSOFT DF32/DF42, type UP
;  board 0 / interface 0
;  one secondary station max.
;  IFRAME maximum 512 bytes
;device = A2C0,DU06,"\\.\DFXX0",0,0,US,0,512,0,0,0,du06us.par
;  via COMSOFT DF32/DF42, type US
;  board 0 / interface 0
;  IFRAME maximum 512 bytes
;device =A2C0,DU06,"\\.\DFXX0",0,0,B,0,512,0,0,0,du06b.par
;  via COMSOFT DF32/DF42, type B
;  board 0 / interface 0
;  IFRAME maximum 512 bytes
;device=A2C0,DU06,"SOCKET","winnt99",10000,B,1,512,30,5,0
;  via SOCKET, type B
;  peer computer WINNT99
;  socket number 10000
;  IFRAME maximum 512 bytes
;device = A2C0,DU06,"SOCKET","",10000,B,1,512,30,5,0
;  via SOCKET, type B
;  peer computer is active
;  socket number 10000
;  IFRAME maximum 512 bytes
;----------------------------------------------------------------------------
; KS100 computer interfacing
;----------------------------------------------------------------------------
;device = 8500,KS100,1,0,0,0
;  operation via first CP1413
;device = 8500,KS100,2,0,0,0
;  operation via second CP1413
;----------------------------------------------------------------------------
; UCP-2 computer interfacing (TCP/IP on board)
;----------------------------------------------------------------------------
;device = C100,UCP-2,0,0,0
;  no parameters
;----------------------------------------------------------------------------
; timer
;----------------------------------------------------------------------------
;device = 8010,ZIG1,"orgzeit"          ; time adjustment relative to Windows time
;device = 8050,ZIG1,"$NT_TIME$"        ; ORG+Windows time synchronization
;device = 8050,ZIG1,"$UTC_TIME$"       ; time relative to GMT
;device = 8010,ZIG2;                   ; time with "millisecond" accuracy
;----------------------------------------------------------------------------
; PE F7 process element
;----------------------------------------------------------------------------
;device = 2000,ALEM,"\\.\PE_F7"
;  no parameters
;  PSF =
;  see manual
;----------------------------------------------------------------------------
; pseudo device for PSD
;----------------------------------------------------------------------------
;device = 8080,FTNT,"pipepsd",0
;  via pipe pipepsd
;  message length 4 kbytes max.
;device = 8080,FTNT,"pipepsd",32
;  via pipe pipepsd
;  message length 32 kbytes max.
;---------------------------------------------------------------------------
; Promea MX
; (parameterize COMx interfaces with setall.bat)
;---------------------------------------------------------------------------
;device = 8202,EAMX,"\\.\COM2"
;  via serial interface
;  no break detection
;---------------------------------------------------------------------------
; ES terminal
;---------------------------------------------------------------------------
;device = 8202,EAES,"\\.\COM2",34
;  via serial interface
;  transparent input
;  transparent output
;  9600 bauds
;device = 8202,EAES,"\\.\COM2",04
;  via serial interface
;  input not transparent
;  output not transparent
;  9600 bauds
;device = 8202,EAES,"\\.\COM2",03
;  via serial interface
;  input not transparent
;  output not transparent
;  4800 bauds
;---------------------------------------------------------------------------
; watchdog
;---------------------------------------------------------------------------
;watchdog = "\\.\bello", 10, 15, 30, 30
;watch = STOP, Reset
;watch = STOP, RBORG
;watch = 8104, RBSIC
;watch = 8008, RBNT
;watch = 8008, RESET
```

9.4 Template of the Parameter File *rpar.sys*

```
;-----------------------------------------------------------------
;  M2000: R-emulation: parameter definition example, formatted 80 characters/line
;     MEDIS compatible (no vowel mutation)
;
;
;  -----basic parameters-----
;  model (R10/R10V/R20/R30)
   CPU=R10V
;
;  main memory size in KW (64..1024)
   maxmem = 1024
;
;  main memory buffering
;  image = "image"
;
;  Hexa switch (boot address)
   BOOT=8025
;
;  maintenance panel (usually not required)
;  WTF
;
;  command transmission to device processes via EA-U
;  EAU
;
;  -----visual display units-----
;  visual display unit to connection point 1, 0 is virtual console
;
;  device = 1,0,3974 ,"COM2"          ; serial connection of ZBE3974
;  device = 1,0,3974R,"COM2"         ; serial connection of ZBE3974R, DS074, DS075, TE2000 AX
;  device = 1,0,3974M,"\\.\COM2"      ; ZBE3974M, DS075 DISIT, TE2000 DX
;
;  device = 1,0,3974RT,"DSSE0"        ; local emulation TE2000 AX, TE-KONS
;  device = 1,0,3974MT,"DSSE0"        ; local emulation  TE2000 DX
;  device = 2,0,3974MNET,"20001"      ; DISIT emulation via network
;  device = 2,1,3974RNET,"20002"      ; TE2000 AX emulation via network
;
;  -----disk occupation mode-----
;  mode = share;                      ; = share, shared disk access;
;                                     ; without mode =, exclusive disk access
;  -----External memory -----
;  device = 5,0,3941 ,"plsk\disk0"
;  device = 5,0,3942 ,"plsk\disk0"
;  device = 5,0,3945A,"plsk\disk0"
;  device = 5,0,3945B,"plsk\disk0"
;  device = 5,0,3945C,"plsk\disk0"
;  device = 5,0,3946 ,"plsk\disk0"
;  device = 5,0,3948 ,"plsk\disk0"
;  device = 5,0,3948A,"plsk\disk0"
;  device = 5,0,3948B,"plsk\disk0"
;  device = 5,0,3949 ,"plsk\disk0"
;  device = 5,0,3949A,"plsk\disk0"
;  device = 5,0,3949B,"plsk\disk0"
;  device = 5,0,3949C,"plsk\disk0"
;  device = 5,2,3948 ,"plsk\plsk2"
;  device = 5,3,3948 ,"plsk\plsk3"
;
;  optical disk OP11 on MEC-R container
;  device = 5,0,OP11 ,"\\.\PhysicalDrive1"
;  MECR = 5
```

```
;
;  removable magnetic disk WP256 on MEC-R container
;  device = 5,0,WP256 ,"\\.\PhysicalDrive2"
;  MECR = 5
;
;  magnetic tape cassette MK82 on MEC-R container
;  device = 5,0,MK82 ,"\\.\Tape0"
;  MECR = 5
;
;  -----timer-----
;  PROMEA timer
;  device = 1,3,ZIG1                      ;ORG time only, 1st time call is discarded
;  device = 1,3,ZIG1,"$NT_TIME$"          ;ORG+Windows time synchronous, 1st time call is discarded
;  device = 1,3,ZIG1,"$SYS_TIME$"         ;absolute time values, 1st time call is discarded
;
;  device = 1,3,ZIG1X                     ; ORG time only, 1st time call is transmitted
;  device = 1,3,ZIG1X,"$NT_TIME$"         ;ORG+Windows time synchronous, 1st time call is transmitted
;  device = 1,3,ZIG1X,"$SYS_TIME$"        ;absolute  time values, 1st time call is transmitted
;
;  timer 3691A
;  device = 1,3,ALE1
;
;  -----pseudo device for PSD / WINCC-PSD-----
;  device = 2,0,FTNT,"ftnt0",0,100,0
;  device = 4,0,FTNTX,"\\server1\pipe\ftnt0",7,200,1
;  device = 2,0,FTNTD,"ftntd0",0,512,0
;  device = 2,3,FTNTDX,"\\server1\pipe\ftntd0",8,512,0
;  device = 2,0,WINCC,"wincc0",8,1024,0
;
;  -----special Promea MX1-----
;  device = 3,0,EAMX1,"\\.\COM13",70,20,0x9
;
;  -----ES terminal-----
;  device = 1,1,EAES,"\\.\COM2",0x40
;
;  -----floppy-----
;  device = 3,0,3944,"\\.\A:"
;
;  -----printers-----
;  printer operation on parallel port
;  specification of crlf replaces LF by CR/LF for alphanumerical output
;  (important for laser printers)
; device = 1,1,DRUA,"LPT2",crlf
;  print output to WinNT file
; device = 1,2,DRUA,"\sicomp\trc\drfile
;  serial printer on serial port
; device = 2,0,DRCOM,"\\.\COM11",3915,,100
;  serial printer on port 11, type 3915
;  write monitoring time 20s
;  delay time 100ms
;  print output to Windows window
; device = 4,0,DRWIN,"DRUA4"
;  print output via Windows printer driver
; device = 4,1,DRSPOOL,"LPT1"
;  print to file, parameter-controlled
; device = 2,1,DRSPOTX,"\sicomp\spotx.par"
;
;  -----interface connections-----
;  DUST3964R via serial interface
; device = 7,0,3964R,"\\.\COM11",011
;  DUST3964R via TCP/IP
; device = 7,0,3964R,"SOCKET",22223,winnt7:22225
; device = 7,0,3964R,"SOCKET",22223,220.250.1.155:22225
;  DUST3965R via DF32/DF42
; device = 7,1,3965R,"\\.\DFXX0",0,0,512
;  DUST3965R via TCP/IP
; device = 7,1,3965R,"SOCKET","WINNT7",22220,512
;  DUST3966 via DF32/42
; device = 7,1,3966,"\\.\DFXX0",0,0,US,0,512,0,0,0,g:\sicomp\3966us.par
;  DUST3966 via TCP/IP
; device = 7,1,3966,"SOCKET","WINNT7",22223,US,1024,0,0,0
;  DUST 3961 via PARIF
; device=1,1,3961,"SOCKET",HOSTNT:22227,HOSTRR:22229,1024,0;
;  ----- KS N16 emulation-----
;
; device = 7,0,CS275,"\\.\NATDEV",300
```

9.4.1 Template of the StartBatch *emu_strt.bat*

```
@echo off

REM M2000 startbatch (user template)

REM Wait for Windows runup to complete
REM wait4                 - waits until all Windows drivers and services are started

REM Set up working directory
REM %m2000disk%           - the variable only applies if the user system environment has been installed
REM cd %m2000dir%\trc     - the variable only applies if the user system environment has been installed
k:                        - set up the user partition
cd \m2000\sicomp\trc      - set up the start directory

REM Set up the serial interfaces
call  ..\setall.bat       - if all devices are operated, this batch must be adapted accordingly

REM Start M-emulator with DOS window
REM start "IPKS_SIC-M" /min sic_m ..\mpar_IPKS.sys – starts the emulation ( M ) with a DOS window (minimized)
                                                  - with the parameters of the config. file mpar_IPKS.sys

REM Start M-emulator without DOS window
sic_m ..\mpar_IPKS.sys                            - starts the emulation without DOS window

REM Start R-emulator with DOS window
REM start "EMULATOR" /min sic_r ..\rpar.sys       - as above, but for ( R )

REM Start R-emulator without DOS window
REM sic_r ..\rpar.sys                             - as above, but for ( R )
```

9.5 M2000 Error Notification Form

IPKS
Prozess-Software-Entwicklungs GmbH
H. Großheimann
Arnold-Dehnen-Straße 48
47138 Duisburg
Fon: 0203-410966-12
Fax: 0203-410966-25

eMail: support@ipks.de

Sender

Company	
Name	
Department	
Address	
Telephone	
FAX	
eMail	
Date	

Version M2000 □ Dongle/serial no. □

SICOMP M Emulation □ Windows Version □
SICOMP R Emulation □ Servicepack Version □

TE2000 AX □ Version □
TE2000 DX □ Version □

M2000 Options □

Have errors already been conveyed by telephone?

Yes □ date: □ to: □

Error description:

The following documentation is provided:
see also: Error Analysis by IPKS GmbH

9.6 Error Analysis by IPKS GmbH

To simplify error analysis when problems with the emulation are experienced, please provide the IPKS system engineer with the following information (preferably stored in zip archives):

1. The parameter files *mpar.sys(M), rpar.sys(R)* and *setall.bat.*

2. The APPLICATION log and SYSTEM log from the Windows Event Viewer.

These logs can be stored in a file. To do so, select

(Windows2000/WindowsXP)
Start–Settings–Control Panel–Administrative Tools–Event Viewer – *Application***/** *System*

(WindowsNT4)
Start– Programs – Administration(General) – Event Viewer-*Application***/** *System*

and select the log file.

(Windows2000/WindowsXP).
On the **Action** menu, click **Save log file as ... .**

(WindowsNT4). Select
Save log file as ...

In the **Save As** dialog box, click the directory to which you want to save the file, and then type a name for the file.

3. Windows diagnosis report (complete).

This report can be stored in a file. To do so, select

(WindowsNT4)
Start–Programs–Administration(General)–Windows NT Diagnosis

On the **File** menu, click **Save report as ...** .
In the **Save As** dialog box, click the directory to which you want to save the file,
and then type a name for the file.

(Windows2000)
Start–Settings–Control Panel–Administrative Tools–Computer Management

Click to expand *System information*. On the **Tools** menu, click
Windows and **Windows Report Tool**. Click **Change System File Selection** -
system information is collected. Click **Select All, Ok and Next**.
In the **Save As** dialog box, click the directory to which you want to save the file, and then type a name for the file.

(WindowsXP)
Start– Programs–Accessories–System Tools–System Information

Click *System overview* and then **File->Save** to save the system info file.

10 Index

3941 26
3942 26
3945 27
3946 27
3947 27
3948 27
3949 27
3974MT 25, 26
3974MTNET 25, 26

ALE1 27
ALE1ONLY 88
ALEM 10, 135

BDE terminal 94
Bootstrap loader 23, 30, 68

Cache 18
Communication
  KS100, CS275, CP1400 56
Communication processor
  UCP-2 57
Computer interface connection 11
Computer interfacing DU03 49
Computer interfacing DU03 8
Computer interfacing DU04 50, 51
Computer interfacing DU05 53
Computer interfacing DU05 com 51
Computer interfacing DU06 50
Computer interfacing DUST3966 86
Computer interfacing DU05 55
Computer interfacing DUST3964R 83
Computer interfacing DUST3964R via TCP/IP 84
Computer interfacing DU02 48
Computer interfacing DU06 54
Computer interfacing DUST3962 84
Computer interfacing DUST3965R 85
Console emulation 7
CP1400 12, 26, 56
CPU type 29
CPU type 67
CPU type 23
CS275 12, 26, 56

DEnet75 12
Device definition 24
Device emulation DS074 95
DF32 9, 12
DF42 9, 11, 12
DFCONS 140
Display unit type 35
Display unit types 71
Dongle 13, 140
DR202 25, 27, 44, 77
DRCOM 24, 25, 27, 45, 78
DRSPOT 25, 41
DRSPOTS 25, 41
DRSPOTX 25, 27, 42, 79
DRUA 25, 27
DS074 26, 33
DS074NET 26, 35
DS075 26, 31
DS075 DISIT 25, 26
DSnet75 12
DU02 48
DU03 49
DU04 9, 26, 50, 51
DU05 9, 26, 51, 53, 55
DU05 device emulation 98
DU06 9, 25, 26, 54, 55
DU06 device emulation 100
DUST 3962 11
  fiber optic interface 84, 87
DUST3964R 11, 27, 83, 84
DUST3965 27
DUST3965R 11, 85
DUST3965R device emulation 108
DUST3966 12, 27, 86, 125
DUST3966 device emulation 111

EAES 26, 27, 66
EAMX 26, 65
EAMX1 93
Emulator core process 5
Starting the emulation 28
ETC M 12

FACO30 140
Floating-point arithmetic 6
Floating-point processor, deactivate 68
FLOP 25, 27, 38, 74, 140
Floppy disk 8, 18, 140
Floppy disk drive 38, 74
FP0xx 38
FP0XX 25
FTNT 60, 90
FTNTD 62, 90
FTNTDX 62, 90

Graphics card 18

Hard disk 7
Hard-Disk 18

Installation 19
Interface concentrator 118

KS N16
  (connection to TELEPERM) 94
KS100 26, 56

Magnetic tape cassette drive 8
Magnetic tape cassette drive 39
Magnetic tape cassette 75
Main memory retrieval file 23, 29, 67
Main memory size 29, 67
Maximum main memory 23
MCSAVE 128
MEC-28R 68
MK82 25, 27, 39, 75
MODE 73
  (disk occupation mode) 73
  (disk occupation mode) 37

Occupation mode, disk drives 73
Occupation mode, disk drives 37
ODBC 26
OP11 75

OP31
  MO Disk 25
OP31 Emulation 120
Optical disk OP11 75
Optical disk OP31 39

PARIF 10
PCCOPY 127
PE F7 10, 60
PE3600 10, 114
Performance 4
Printer control characters 140
Printer operation, (parallel) 76
Printer operation (parallel) 40
Printer operation (serial) 77
Printer operation (serial) 44, 45, 78
Printer output to Windows file 41
Printer output to Windows file   77
Printer output to Windows window   46
Printer output to Windows window 81
Printer output via Windows printer driver 76
Printer output
  via Windows printer driver 40
Printers 7, 18, 140
Process element PE F7 60
Processor 18
WAIT4.EXE program 129
PROMEA MX 11
Promea timer 59, 88
PROMEA1 11
PS048 25
PS048, PS049 37
PS049 25
Pseudo 63, 89, 91

Range of services 6
DU05 computer interfacing 51
Removable magnetic disk WP256 75

Scope of delivery M 16
Scope of delivery M2000 14
Scope of delivery R 16
SCU service
  SwitchContrlUnit 126
Serial interfaces 18
SicView 131
SOLDIS
  MO Disk Copy 130
SPOOL75 12
Stopping the emulation 28
Streamers 18

TE2000 AX via network 71
TE2000 DX via network 35, 70, 71
TE2000 DX via network 34
TE2000 DX via network 34, 70
Terminal 18
Terminal emulation, local 70
Terminal emulation, local 33
Time behaviour 123, 125
Timer 8, 123, 125
Timer 3691A 88
TrcView 141

UCP-2 57

VDU2000 25
VICOM 123
Video terminal emulation 6
Video terminals 6
Virtual console 23, 30
Visual display units, serial connection 69
Visual display units, serial 31

WINCC 63
Windows
  Win2000/WinXP 18
Windows time 59

ZE (central unit) emulation 16
ZIG 59
ZIG1 25, 27